

AD-A144 537

A METHOD OF PHOTO DATA REDUCTION WITH DESIGN
CONSIDERATIONS FOR THE NOVA. (U) NAVAL BIODYNAMICS LAB
NEW ORLEANS LA J J LAMBERT JAN 84 NBDL-84R001

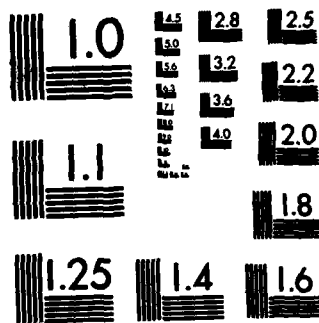
1/1

UNCLASSIFIED

F/G 9/2

NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NBDL - 84R001

12

A METHOD OF PHOTO DATA REDUCTION, WITH DESIGN CONSIDERATIONS
FOR THE NOVA 800® AND UNIVAC 1100/83® COMPUTERS

James J. Lambert

AD-A144 537



January 1984

DTIC
ELECTE
AUG 21 1984
S B

NAVAL BIODYNAMICS LABORATORY
New Orleans, Louisiana

Approved for public release. Distribution unlimited.

84 08 21 029

CTIC FILE COPY

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NBDL-84R001	2. GOVT ACCESSION NO. AD-A144537	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Method of Photo Data Reduction, with Design Considerations for the NOVA 800® and UNIVAC 1100/83® Computers		5. TYPE OF REPORT & PERIOD COVERED Research Report
7. AUTHOR(s) James J. Lambert		6. PERFORMING ORG. REPORT NUMBER NBDL-84R001
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Biodynamics Laboratory P. O. Box 29407 New Orleans, LA 70189		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Medical Research & Development Command Bethesda, MD 20814		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS M0097PN001-5001 63766N
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE January 1984
		13. NUMBER OF PAGES 86
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) data analysis, computer program, photo data reduction, photo digitizing system (PDS), high speed camera, biodynamic experiments, inertial environ- ments, impact injury models		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Digitized photo data acquired during impact experiments must be scaled and converted to a format suitable for analysis. Once converted, this data requires comprehensive graphical presentation for efficient interpretation. This report presents a detailed description of the software developed to accomplish such tasks in both a production and an interactive environment. Procedures utilizing the design presented have been instituted at NBDL and found to be effective.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

NBDL - 84R001

A METHOD OF PHOTO DATA REDUCTION, WITH DESIGN CONSIDERATIONS
FOR THE NOVA 800® AND UNIVAC 1100/83® COMPUTERS

James J. Lambert

January 1984

Naval Medical Research and Development Command
Research Work Unit No. MO097PN001-5001

Approved by

Released by

J. C. Guignard
Chairman, Editorial Review Board

Captain L. E. WILLIAMS MC USN
Commanding Officer

Naval Biodynamics Laboratory
P. O. Box 29407
New Orleans, LA 70189

Opinions or conclusions contained in this report are those of the author(s) and do not necessarily reflect the views or the endorsement of the Department of the Navy. Approved for public release; distribution unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government.

SUMMARY PAGE

THE PROBLEM

Digitized photo data acquired during impact experiments must be scaled and converted to a format suitable for analysis. Once converted, these data require comprehensive graphical presentation for efficient interpretation. This report presents a detailed description of the software developed to accomplish such tasks in both a production and an interactive environment.

FINDINGS

Procedures utilizing the design presented have been instituted at NBDL and found to be effective.

RECOMMENDATIONS

In an environment requiring data reduction processing of photo data, the design presented herein should be considered.

ACKNOWLEDGEMENT

This research was sponsored by the Naval Medical Research and Development Command and was performed under Navy Work Unit No. M0097PN001-5001. The author acknowledges Mr. William R. Anderson for valuable technical collaboration proving useful in the presentation of findings of this research. Miss Judy B. Johnson expended much time and patience in creating and maintaining the report text using NBDL word processing equipment. Additionally, Mr. Arthur M. Prell and Miss Margaret M. Harbeson were responsible for the creation of photographic reproductions (of illustrations) and the manuscript editing functions, respectively.

Trade names of materials or products of commercial or nongovernment organizations are cited only where essential for precision in describing research procedures or evaluation of results. Their use does not constitute official endorsement or approval of the use of such commercial hardware or software.

TABLE OF CONTENTS

	<u>PAGE</u>
INTRODUCTION.....	1
1. PHOTO DATA CONVERSION PROGRAM.....	2
A. INTRODUCTION.....	2
B. DISCUSSION.....	2
C. MAIN PROGRAM AND SUBROUTINE DESCRIPTIONS.....	5
2. PLOT PROGRAM FOR PHOTO DATA.....	6
A. INTRODUCTION.....	6
B. DISCUSSION.....	6
C. MAIN PROGRAM AND SUBROUTINE DESCRIPTIONS.....	9
3. REFORMATTING (SIGN CHANGE) PROGRAM FOR PHOTO DATA.....	10
A. INTRODUCTION.....	10
B. DISCUSSION.....	10
C. MAIN PROGRAM DESCRIPTION.....	11
4. OPERATION OF PHOTO DATA REDUCTION SYSTEM.....	11
A. INTRODUCTION.....	11
B. DISCUSSION.....	11
C. PRODUCTION RUNSTREAM.....	15
CONCLUSION.....	17
FIGURES.....	18
REFERENCES.....	39

APPENDICES

DTIC
ELECTE
S **D**
AUG 21 1984
B

111



Unannounced Justification		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
By _____		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A-1		

A METHOD OF PHOTO DATA REDUCTION, WITH DESIGN CONSIDERATIONS FOR THE NOVA 800® AND UNIVAC 1100/83® COMPUTERS

INTRODUCTION

The Naval Biodynamics Laboratory (NBDL), located in New Orleans, LA, is an internationally recognized laboratory which performs experimental research to determine the effects of aircraft crashes, ship motion, vibration, aircraft ejection and parachute opening forces on the health and performance of Navy personnel. On-going research programs use high speed instrumentation cameras to record the motion of test subjects during biodynamic experiments. The films are digitized and the 3-dimensional motion is reconstructed and analyzed.

The procedures and programs used to convert the digitized photo data into a format compatible with a large-scale general purpose computer (UNIVAC 1100®), to plot the data for identification of data errors, and to prepare the data for subsequent processing have recently undergone major revision. The objectives of the revision were to:

1. Modify existing software to be compatible with announced changes in the UNIVAC operating system.
2. Use new graphics support capabilities to improve error detection procedures, reduce turn-around time of analysis, and produce output compatible with all available graphics devices.
3. Improve efficiency of operations personnel.
4. Standardize data formats and I/O access methods to be compatible with existing archival data.
5. Develop a modular organization of software which provides greater operational flexibility.

This report documents the new procedures and programs.

DESCRIPTION OF SOFTWARE

Preparation of digitized photo data for subsequent reduction and analysis requires a process consisting of several tasks performed under computer control. Software supporting each task conforms to the principles of modular design so that task sequences may be selected to meet specific needs of the data being processed.

In order to process digitized photo data through a data reduction flow, major tasks are selected and merged in a single runstream. The three available programs are:

1. Convert photo data from PDS (Photo Digitizing System) NOVA 800® format to UNIVAC 1100/83® format for subsequent analysis.

2. Produce film frame plots and x-y trajectory plots of target data in order to provide graphical presentation for error analysis.

3. Convert data digitized manually into the format of data tracked under computer control.

Each of these components is documented in detail in the body of this presentation. It is assumed that the reader is familiar with the operating system of the UNIVAC 1100/83[®] computer (see reference 3) and with the UNIVAC ASCII FORTRAN programming language (see reference 4). Although the software presented herein is intended for specific computers, plotting devices and software support packages, it has been designed to minimize the conversion effort for other systems.

1. Photo Data Conversion Program

A. Introduction. This section describes a UNIVAC 1100/83[®] program which converts photo data from PDS (Photo Digitizing System) data tapes to UNIVAC 1100/83[®] format. In converting data to a format compatible with a large-scale general purpose computer (UNIVAC 1100/83[®]), the photo data conversion program provides a suitable environment for efficient data reduction and graphical presentation.

B. Discussion. Input to the conversion program is a PDS data tape. It contains digitized photographic data from at least one experiment, or run. For a particular run, as many as three cameras are used to photograph targets on anatomical mounts strapped to the experimental subject. Each reel of film generated from these cameras during the run is digitized (on the NOVA 800[®] system), with the output being written to a PDS data tape. Data for several runs may be written to a single PDS tape. This tape contains four record types. The first is a header record, which contains identification information for film from a particular camera used during a run. This record includes such items as run number, PDS record type, camera site number (1, 2 or 3), julian date the film was digitized and time-of-day for the first film frame (all character data on the PDS tape is in ASCII format and numeric data is in twos complement integer format). The second record type is the time record, which identifies time-of-day for each film frame, or data point. The time data values originate from LED displays on the film frames themselves. There are 250 data points on a time record (the first point being the time at first motion of the subject). Identification information for the reel of film the time record describes is also included in the record, i.e., run number, PDS record type and camera site number. The final two record types are the X and Y data records, respectively. Each contains 250 data points measuring film plane displacement (inches) from a stationary origin on the film. An X and Y data record exists for every photo target tracked during the run. Each data record contains (in addition to displacement values) run number, PDS record type, camera site number and a target number. The target number contains information describing the anatomical mount type on which it is located. For instance, there are head mounts, neck mounts, mouth mounts and pelvic mounts - all designed to track motion of a particular segment of the body.

Data on a PDS output tape is organized by reels of film. That is, all data for a particular reel of film (with its unique header, time and data records) are grouped together into a file. The order in which the records appear is always the same: header record, time record, target 1 (x values), target 1 (y values) ... target N (x values), target N (y values), where N = number targets digitized on film. In some experiments more than the standard 250 frames are digitized. In this instance the data records for the extra frames (always a multiple of 250) are written to the PDS tape immediately after the final data record containing data for the previous 250 frames. An end-of-file separates each group of data records for a particular set of 250 frames. A double end-of-file signals the end of data on the tape.

The conversion program reads the NOVA 800® input tape and stores it in an input buffer. The PDS characters are in 8-bit ASCII format and the numeric data is either in 16-bit or 32-bit integer format. The conversion program determines the format of the information to be converted from the format specifications of the PDS input tape (see Figures 1b-1d). All character data is converted from 8-bit NOVA ASCII to 9-bit UNIVAC ASCII format. Numeric data is either converted to 36-bit UNIVAC floating point format or 36-bit UNIVAC integer format, depending on the format type of the information being converted. For instance, PDS record type is converted from NOVA 16-bit integer to UNIVAC 36-bit integer, while displacement (data) values are converted from NOVA 32-bit integer to UNIVAC 36-bit floating point format. All converted displacement values are then scaled from inches to meters. This is done to provide data compatibility with a group of data analysis programs collectively called EASYFLOW*.

The converted and scaled photo data must be output in a standard file format suitable to subsequent processing; in addition this data must be output in FIELDATA format, which is the NBDL archival standard. This second requirement presents difficulties in an ASCII FORTRAN programming environment. In order to provide for both needs an I/O access package (see Reference 1) has been developed to (1) provide standard I/O access methods for an established photo data file format and (2) maintain data format compatibility with the NBDL archives. The I/O package provides directory maintenance for the user. The conversion program utilizes this capability to output the standardized direct access record. This directory contains unique run identification numbers (for all runs on the output tape) and accompanying sector locations for both header and time records for the run. The directory provides means for rapid direct access of photo data for a given run number.

* EASYFLOW is an NBDL term representing a data reduction and analysis system requiring input data of a specific format. EASYFLOW uses converted photo data to derive position coordinates in 3-dimensional space of targets affixed to anatomical mounts (which are strapped on to the experimental subject). These data are subsequently used to compute and plot displacement, linear (and angular) velocity and linear (and angular) acceleration of the subjects.

There are four record types, in addition to the directory, which the conversion program outputs to a UNIVAC mass storage file. The first of these is the header record. There exists a unique header record for each camera site used in a run. There are no more than three and no less than two cameras used in a run, each assigned a site identification number of 1, 2 or 3. This site ID and the run number in concert make the header record unique. A keyword is stored in the header record along with identification information such as the run number and site ID. The keyword identifies the record as a header. The second type of output record is the time record. It contains the time-of-day information for each of 250 film frames. In addition the time record includes the run number, the site ID and a keyword for identification. The third record type is a photo data record. It contains displacement values (from a fixed point on the film plane) in meters of a particular photo target on the reel of film. Also included are the run number and site ID of the film, along with a keyword identifying the record type. This keyword also contains information identifying the anatomical mount (on which the photo target resides) and the axis type of the data (X or Y). The combination of header record, time record and data records (for all targets on the reel of film) is written to disk, in the order listed, by the conversion program. Data for each reel of film with a camera site ID of 1 are grouped together on the output file and separated by double ends-of-file*. An additional end-of-file is written after the final data record of the final reel of film for site ID 1. This signifies end-of-data. However, the direct access table follows this final end-of-file. It is located at the end of the file so as not to limit its size. In order to provide access to the directory a "locator table" is written as the initial output record. It contains the sector address of the directory.

The above output file structure is duplicated for camera site 2 (and 3 if necessary). A separate UNIVAC disk file is thus created for each site ID. Each of these files is in a standardized format, and serves as input to a UNIVAC runstream designed to write them to a UNIVAC 1100/83® data tape.

The conversion program is designed to correct three common situations in which PDS operator error has occurred. These three special cases are:

- (1) The run number of a header record for a reel of film is entered incorrectly by the PDS operator and written to the PDS data tape.
- (2) A reel of film is re-digitized by the PDS operator and output to the end of the same PDS data tape as data from the original reel of film.
- (3) The camera site ID of a header record for a reel of film is entered incorrectly by the PDS operator and written to the PDS data tape.

* If more than 250 frames of data were digitized on a reel of film, each group of data records for the extra frames (always a multiple of 250) is written in succession and separated by single end-of-file.

In cases (1) and (3) the conversion program uses correction fields in card image input in order to scan the header record for the incorrect entry and then correct it. This is done prior to writing the header to disk. In case (2) the conversion program scans for the first occurrence of the data file for a reel of film. Once found, this data file (header, time and data records) is bypassed and thus not converted and output. However, the second occurrence of a data file for the reel of film is converted and output in the usual manner. The photo conversion program accomplishes the following objectives:

(1) Converts photo data to a format compatible with a large-scale general purpose computer (UNIVAC 1100/83®).

(2) Scales displacement data values from inches to meters in order to provide data compatibility for subsequent processing.

(3) Provides for correction of common PDS operator errors.

(4) Accomplishes output of data with standardized I/O access methods.

(5) Maintains data format compatibility with the NBDL archives.

C. Main Program and Subroutine Descriptions:

(1) MAIN (see Appendix 1b for listing) - This is the driver program which controls all program operations. All correction cards are read by this program and used to rectify the three special cases earlier noted. MAIN then calls subroutine CONVT in order to read and convert the first PDS input record (see Figures 1e-1g for input record formats). The rest of the input records (non-header) for the run are read and converted by virtue of calls to subroutine CNVT1. Here the converted records are written to mass storage. Once a run has finished processing, control returns to MAIN and the process repeats itself until all runs have been processed. At this point a triple end-of-file, denoting end-of-data, is written to the output file.

(2) CNVT1 (see Appendix 1c for listing) - This subroutine controls the conversion and output of all data for each camera. The main program reads the header records and transfers control to this subroutine for reading and conversion of the position and time records. The position data is converted from inches to meters*. Each header, time and position record is written to mass storage by a call to the photo IO subroutine PUTLCD (see Appendix 4c). Data for each site ID is written to a separate mass storage unit (see Figure 2a). As mentioned earlier, the necessary data directories are maintained and written by the photo IO subroutines. After processing an entire run, CNVT1 writes a double end-of-file for each output unit and returns control to the calling program.**

* During the scaling process all data values of 0.0 (points not digitized) are set to 999.0

** If more than 250 frames were digitized for each target in a run, only a single EOF (End-of-file) is written after the final data record for each set of 250 frames.

(3) CONVT (see Appendix 1d for listing) - This subroutine is called by CNVT1 to perform the actual conversion of data from PDS format to UNIVAC 1100/83® format. It is called after each read of a PDS input record. All character data is converted from 8-bit ASCII to 9-bit ASCII format. Numeric data is converted from 32-bit binary integer to UNIVAC 1100/83® 36-bit binary floating point format (see Figures 2b-2f for output record formats).

In this subroutine the sign bit is extended for negative numbers; these negative values are data entered manually by the PDS operator by using the X,Y crosshair input.

(4) SPHAVG (see Appendix 1e for listing) - This subroutine calculates the X and Y position of a stationary reference target relative to the film sprocket hole for each camera used in a run. Sprocket hole values for five consecutive frames are averaged, the result being subtracted from the averaged sled (chair) displacements for the same five frames in order to produce the final value. The X and Y sprocket hole averages are printed for each camera and in those cases where there is no sprocket hole information on the PDS tape, the values can be read from the film by the PDS operator and calculated by hand. Execution of EASYFLOW is expedited by providing sprocket hole averages at the time of conversion. This information is required for an internal consistency check on the data collection and system.

(5) ASC8T9 (see Appendix 1f for listing) - This subroutine converts a string of 8-bit ASCII characters to a UNIVAC 1100/83® compatible 9-bit ASCII character string. All character data read from the PDS input tape is converted by the subroutine. The high order bit of a 9-bit ASCII character is always off, this bit being the only difference between an 8-bit ASCII character and a UNIVAC 1100/82 9-bit character.

2. Plot Program for Photo Data

A. Introduction. This section describes an ASCII FORTRAN UNIVAC 1100/83® program which plots UNIVAC-compatible photo data as (a) X-Y film frame plots and (b) X-Y trajectory plots. These plots provide a method for error detection of faulty photo data, allowing for correction of the data before any further processing takes place.

B. Discussion. The plot program generates both x-y film frame plots and x-y trajectory plots. The film frame plots (see Figure 5a) plot the position of each photo target (on a reel of film) relative to the film frame origin. Each individual frame of film is plotted, and all target positions in the frame are connected by straight lines; this aids in visual recognition of relative changes in target positions from frame to frame. There is sometimes more than one anatomical mount (see Figure 6 for mount example) visible to a camera during a run. It is for the reason that a separate set of film frame plots are generated for each anatomical mount found on a reel of film. For instance, if two mounts are visible to a particular camera and only one mount is visible to a second camera, three sets of film frame plots are generated for the experiment. A total of 250 film frames are plotted for each set,

which consists of ten pages of twenty-five plots each. On top of each page the plot scale (meters) is listed; this measures the greatest target displacement from the first to last frame on the page. In addition all target positions which were manually digitized (not automatically tracked) by the PDS operator are circled for identification. The film frame plots are used to identify errors which may exist in the data. For instance, a target may have been identified incorrectly by the PDS operator; this problem is visually exposed in the film frame plots. Subsequently the data file is corrected.

The x-y trajectory plots (see Figure 5b) identify the path each target (on a reel of film) travels in the film plane during a run. Each individual point on the plot measures target displacement from a fixed film frame origin. A total of 250 data points are plotted, with the first coinciding with time of first motion of the experimental subject. Since more than one anatomical mount may appear on a reel of film, a separate x-y trajectory plot is generated for each mount on the film. The path of every photo target on a mount (visible to the camera) is plotted, each identified by its own target number. The x-y trajectory plot provides at a glance a visual presentation of target movement (for a given mount) for the entire run duration; this allows for quick detection of data discontinuities. For instance, if the pattern recognition algorithms (used in the photo digitizing process) found the wrong target, a detectable discontinuity results in the x-y trajectory plot.

The plot program for photo data possesses versatile plot output capabilities, made available through the use of DISSPLA® plot software (a product of ISSCO*). All plot commands are output to a catalogued UNIVAC disk file (via DISSPLA® software). This "compressed" plot file can then be used as input to a variety of DISSPLA® post-processor programs, which "decompress" the plot commands and direct them to a plotting device. This disk file may be utilized several times on a variety of plotting devices without once repeating execution of the actual plot program. DISSPLA® post-processor programs exist for such plotting devices as the TEKTRONIX 4014®, FR80® and HP 7220® (all available to NBDL).

There are advantages in using any of these post-processors. For instance, the FR80® post-processor is used in a production environment to generate microfiche and high quality hardcopy. The TEKTRONIX 4014® post-processor can be used to view selected plots on the TEKTRONIX 4014® scope immediately after execution of the photo data plot program. This quick turnaround is of benefit in expedient data analysis. The FR80® is not located on-site, so in order to combine the benefits of quick turnaround and high quality hardcopy an HP 7220® plotter is to be installed on-site. This plotter in addition affords a multi-color plotting capability.

* Integrated Software Systems Corporation

The input to the plot program is UNIVAC-compatible photo data. Character data is in 9-bit UNIVAC ASCII format and numeric data is either in 36-bit UNIVAC integer format or 36-bit UNIVAC floating point format (depending on format specifications). As noted in section 1.8 data is organized on the input files by film reel, with a separate input file for each camera site. A directory exists in each input file, containing the run numbers (each run number identifies a unique film reel) and accompanying sector locations of the header and time records. The data records for a run (or film reel) immediately follow the time record, and a "locator table" for the directory precedes all other records on the file. Input is accomplished with a general purpose I/O package (see Reference 1) which reads FIELDATA (NBDL archival standard) in an ASCII FORTRAN programming environment. Plots are output for each input file (or camera site). Within each file, plots are output by run number (or film reel) and within each run plots are output by anatomical mount.

The PDS operator is responsible for entering time-of-day values (LED displays) for each frame of digitized film. To insure their legitimacy the plot program performs tolerance checks for the time values. The results (see Figure 5c), called a time record analysis, are output to a line printer*.

The plot program for photo data accomplishes the following:

- (1) Provides error detection methods for photo data by generating:
 - (a) x-y film frame plots.
 - (b) x-y trajectory plots.
- (2) Performs confidence checks for time-of-day values for film frames and prints the results.
- (3) Provides plot flexibility features through the use of post-processor programs.
 - (a) Plots can be generated repetitively without repeat execution of the plot program.
 - (b) Plots can be viewed immediately upon execution on a demand terminal scope.
 - (c) Quality hardcopy and microfiche of plots are generated.
- (4) Accomplishes input of data with standardized I/O access methods.

* This analysis may be output to a plotting device, but it is slower and more expensive to do so.

C. Main Program and Subroutine Descriptions:

(1) PLOT (see Appendix 2b for listing) - This is the driver program which controls all program operations. A single input card identifies (a) the number of input files, (b) the time record analysis option, (c) the X-Y contour plot option, (d) the frame plot option, (e) the frame plot format option, (f) the lowest run number to be processed and (g) the highest run number to be processed (see Appendix 2a for card format). Once the input card is read, DISSPLA® is called to initialize the compressor. Plotting is accomplished by virtue of a loop which controls the input and subsequent plotting of photo data for each combination of run/camera site/anatomical mount (see Figures 2b-2f for input record formats).

(2) TIMER (see Appendix 2c for listing) - This subroutine generates a time record analysis for a given run number and site ID. The time record is input via the photo data I/O program (see reference 1). A time interval between frames is approximated by averaging total elapsed time over the 250 frames of data. This increment is used to calculate an "expected time" for each of 250 frames. These times are compared to the actual time record data points in order to test for excessive error (a difference between actual and "expected" time greater than .001 second). Any such discrepancies are annotated with an asterisk (*) on the time record analysis (see Figure 5c).

(3) LDATA (see Appendix 2d for listing) - This subroutine loads photo displacement data into common for access by plot subroutines. This routine is called once for each combination of run/camera/mount. The photo data I/O package is used to accomplish input (in order to read FIELDATA within an ASCII FORTRAN program).

(4) MIXY (see Appendix 2e for listing) - This subroutine determines minimum and maximum X and Y values for a page (twenty-five frames) of frame plots. These values are placed in common for scaling use by the plot subroutines. The minimum and maximum values are selected so that the abscissa and ordinate axes are equal lengths in data units (meters). This allows the viewer of the plots to determine at a glance in which direction the targets were moving fastest. For example, if the targets moved from the far left of the initial frame on the page to the far right of the final (25th) frame, the X filmplane displacement was changing more rapidly than the Y displacement.

(5) PLOTD (see Appendix 2f for listing) - This subroutine consists of two entry points. All plotting in both entry points is done with DISSPLA® software.

A. PLOT1 - This entry point plots the title information for each page of frame plots. Included in the title information are the (i) run number, (ii) camera number, (iii) plot scale (meters), (iv) anatomical mount type and (v) page number. All plotting is done with DISSPLA® software.

B. PLOT2 - This entry point plots all frame plots. It is called once for every frame of photo data to be plotted. Two-hundred fifty frames are plotted for each mount, twenty-five frames/page, ten pages/mount. The target number for all targets tracked (by the PDS operator) in a particular frame are printed directly at their digitized position. Those targets digitized manually are circled for identification. All targets in a particular frame are connected by vectors in order to present the image of motion as the targets begin to change position from frame to frame. Targets may have been deleted or added as the frames advance. Deleted target numbers are printed in the bottom right corner of the frame in which the target was dropped. Added targets are printed in the bottom left corner of the frame in which the target was added. If chair data are available for the current run/camera, a 'C' is printed in the top left corner of each frame. The frame number is printed in the top right corner of the frame.

(6) XYPLOT (see Appendix 2g for listing) - This subroutine plots X vs. Y filmplane displacement values for a given run/camera/anatomical mount. All targets digitized for the mount are plotted on the same graph. A maximum of 250 data points are plotted (and connected) for each target, with the target number appearing coincident with the first point for a curve. The maximum and minimum values of the abscissa and ordinate are printed on the plot. In addition, titling information is listed which includes (i) run number, (ii) camera number, (iii) anatomical mount type and (iv) current date. The X-Y contour plot affords the viewer a quick method of isolating data incongruities.

3. Reformatting (sign change) Program for Photo Data

A. Introduction. This section describes an ASCII FORTRAN UNIVAC 1100/83⁰ program which converts UNIVAC-compatible photo data, digitized manually, into the format of data tracked under computer control. This prepares the data for subsequent computation of 3-dimensional trajectories of test subjects.

B. Discussion. This program is designed to read all target position data for a given range of run numbers and scan for negative data values. Each negative value represents an entry made manually by the PDS (Photo Digitizing System) operator using the x-y crosshairs. All such values are multiplied by -1 and rewritten to the data file. An archival data tape is later generated (with the reformatted data records) by an independent runstream. This tape serves as input to the EASYFLOW runstream.

Input to the reformatting program is UNIVAC-compatible photo data. Character data are in 9-bit UNIVAC ASCII format and numeric data are either in 36-bit UNIVAC integer format or 36-bit UNIVAC floating point format (depending on format specifications). Data are organized on the input files by film reel (or run number), with a separate input file for each camera site (see section 1.8 for a more detailed description). Direct access tables accompany each input file. Input/output is accomplished with a general purpose I/O package (reference 1) which reads/writes FIELDATA (NBDL archival standard) within the context of an ASCII FORTRAN programming environment. Output file formats for the reformatting program are identical to the input file formats, the only difference being that all values in the output data records are positive.

The reformatting program accomplishes the following:

(1) Prepares UNIVAC-compatible photo data for subsequent computation of three-dimensional trajectories (of test subjects) by converting manually digitized photo data into the format of computer-tracked photo data.

(2) Accomplishes input/output of data with standardized I/O access methods.

C. Main Program Description:

SGFLIP (see Appendix 3b for listing) - This is the driver program which controls all program operations. The only subroutines utilized are those belonging to the photo data ASCII FORTRAN I/O program. A single input card identifies (a) the number of input files, (b) the lowest run number to be processed and (c) the highest run number to be processed (see Appendix 3a for card format). The number of input files and the run number range (from the input card) are used to index a loop which systematically (a) reads data records, (b) scans them for negative data points and (c) changes negative values to positive values. Once a data record is completely processed it is rewritten to the input file unless no negative values were found (see Figures 2b-2f for input/output record formats).

4. Operation of Photo Data Reduction System

A. Introduction. This section describes the actual UNIVAC 1100/83⁰ production runstream which controls data reduction processing of NBDL photo data prior to its input to the EASYFLOW runstream.

B. Discussion. As previously noted, there are three main processing steps in the photo data reduction process: conversion, plotting and reformatting. Each step performs an integral function independent of the other steps. This modular organization of steps provides several benefits:

(1) Any step can be performed independently from the others; i.e., each step has stand-alone capability.

(2) Modifications to steps can be effected with:

(a) Complete non-interference with other steps.

(b) Expediency in that all necessary changes take place in a single module.

(3) Input/output may be examined at any point between steps.

(4) Additional steps may be added to the overall process without altering any existing steps, thus providing a desirable environment for enhancements to the system.

Each step, or module, shall heretofore be referred to as an element. Element is a UNIVAC term describing a collection of computer commands. When it is desired to execute an element, or step, the computer is instructed to carry out all commands in that element. The production runstream described in this section executes a collection of three such elements:

- (1) XQTCVT is the conversion element.
- (2) XQTPLT is the plot element.
- (3) XQTSNG is the reformatting (sign change) element.

The first element to be executed is the conversion element XQTCVT (see Appendix 1a). The main function of this element is to execute the photo conversion program. Additionally, the output of the conversion program is written to a UNIVAC data tape. In sequence, the conversion element (1) copies a PDS data tape onto disk, (2) executes the photo conversion program and (3) writes the output (from the conversion program) to a UNIVAC data tape. The UNIVAC tape is saved for 10 days as a precautionary measure. The key output from the conversion element is the converted photo data written to disk by the conversion program.

If it is necessary to correct for PDS operator error (see section 1.8.) the conversion element is altered to include correction options. These options are input to the element in card image via the UNIVAC EDIT processor (see reference 5). The EDIT processor commands are part of the production runstream. Thus the runstream actually edits the conversion element before executing it. The correction options allow for rectification (by the conversion program) of instances where the PDS operator:

- (1) Enters incorrect run numbers on PDS header records.
- (2) Outputs data to a PDS tape for two film reels with the same run number and camera site ID (on the header record).
- (3) Enters incorrect camera site ID's on PDS header records.

The first card image of options (input with the EDIT processor) indicates the number of film reels to be corrected for each of the three categories of problem cases. A card image then follows for each occurrence of a problem case (1); it specifies the run number and camera site ID of (the header record for) the problem film reel as well as the occurrence (of this unique combination) to be corrected. It also specifies the correct run number to replace the incorrect field in the header record. After the last correction card image for case (1), card images follow for case (2). Each of these specifies the run number and camera site ID of the problem reel. The first occurrence of this combination of run number and site ID (in a PDS header record) is bypassed and only the second occurrence (of PDS records with this header identification) is converted to UNIVAC format. Lastly, the correction card images for case (3) follow. Each specifies the run number, camera site ID and

occurrence index for the problem reel. Additionally, it contains the correct camera site ID. If no corrections are necessary on the PDS input tape, only one card image is input. It indicates quantity zero for each of the three problem cases.

The EDIT processor not only controls input of all correction options to the conversion element, but it also is used to enter the reel number of the PDS data tape to be used as input to the photo conversion program. As with the EDIT commands for correction options, these commands also constitute a portion of the overall production runstream.

The second element of the runstream to be executed is the plot element XQTPLT. Its primary function is to execute the plot program for photo data. Additionally, the disk file of DISSPLA® plot commands (output from the plot program) is used as input to the FR80® post-processor program to produce production quality microfiche and hardcopy of all plots. In sequence, the plot element (1) executes the plot program for photo data (the plot program uses photo data residing on disk as input), (2) executes the FR80® post-processor program and (3) writes the disk file of plot commands to a catalogued disk file for public access. As previously noted the plot program uses output from the conversion program as its input. This input consists of disk files of photo data, one for each camera site (1, 2 or 3) used in a series of runs. Output from the plot program, a disk file of DISSPLA® plot commands, serves as input to the FR80® post-processor. The post-processor interprets the plot commands in order to produce microfiche and hardcopy of all plots generated by the plot program (these include the x-y film frame plots and x-y trajectory plots). These plots are used in error detection analysis and are maintained on-site.

Certain plot options exist and are input to the plot element via the UNIVAC EDIT processor. The EDIT commands constitute part of the production runstream (see Figure 8); they are used to edit the plot element before executing it. A single card image contains all the plot options. These options include:

- (1) Number of input files.
- (2) Time record analysis option.
 - (a) Output to line printer.
 - (b) Output DISSPLA® plot commands to disk file.
 - (c) Do not output to any device.
- (3) X-Y trajectory plot option.
 - (a) Output DISSPLA® plot commands to disk file.
 - (b) Do not output to any device.

(4) X-Y film frame plot option.

(a) Output DISSPLA® commands to disk file.

(b) Do not output to any device.

(5) Plot direction option for x-y film frame plots*.

(a) Plot five columns to a page with frame number increasing in each column (plot direction always downward).

(b) Plot five columns to a page with frame number increasing in columns 1, 3 and 5 (plot direction downward) and decreasing in columns 2 and 4 (plot direction upward).

(6) Lowest run number to be plotted

(7) Highest run number to be plotted

In an input file each run number identifies the header record for a single film reel; photo data is plotted for this range of run numbers (or film reels).

Once the plot program has been executed, the plot element executes the FR80® post-processor program to produce microfiche and hardcopy of all generated plots. The plot (disk) file is then copied to a catalogued disk file.

The third and final element of the production runstream is the reformatting element. The primary function of this element is execution of the reformatting (sign change) program. The element also generates a UNIVAC data tape containing output from the reformatting program. This data tape is saved for 180 days and is used as input to the EASYFLOW data reduction and analysis system. In sequence, the reformatting element (1) executes the reformatting program and (2) saves the output on a UNIVAC data tape. The reformatting program uses photo data residing on disk (output from the conversion program) as input. These same files, after being reformatted to reflect automatically digitized photo data, are output to the UNIVAC data tape.

The UNIVAC EDIT processor is utilized (in the production runstream) to input (1) reformatting program options and (2) the output tape file name. The input options include:

(1) Number of input files.

(2) Lowest run number to be processed

(3) Highest run number to be processed

In an input file each run number identifies the header record for a single film reel; photo data is reformatted for this range of run numbers (or film reels).

* This option is used only in the event x-y film frame plots are generated.

C. Production Runstream. The photo data reduction runstream is a collection of computer commands designed to execute the three steps (elements) in the data reduction process. The runstream consists of (1) UNIVAC Operating System commands (see reference 3) and (2) UNIVAC EDIT processor commands (see reference 5). All three elements to be executed are stored in a single program file. The runstream (1) copies the program file onto a scratch disk file, (2) enters the input options for each element and (3) executes each element. The only input to the runstream is a single PDS data tape containing digitized photo data. Output from the runstream includes:

- (1) A 10-day UNIVAC data tape of converted photo data.
- (2) Catalogued disk file containing DISSPLA® plot commands for all generated plots.
- (3) Microfiche and hardcopy of all generated plots.
- (4) A 180-day UNIVAC data tape of converted and reformatted photo data.
- (5) Line printer listing containing:
 - (a) Time record analyses.
 - (b) Operating system diagnostics.
 - (c) User-generated diagnostics.
 - (d) Operating system accounting information.

Several considerations entered into the design of the photo data production runstream. The more significant of these are noted as follows:

(1) In storing plot commands on a catalogued disk file, plots can be generated as many times as necessary (and on several plotting devices) without once repeating execution of the actual plot program.

(2) In some instances (a film reel of) photo data is redigitized in order to improve data quality. It is important (in computing 3-dimensional position of the test subject) that a minimum total of three targets (not all from the same camera site) are digitized successfully for a given test run; these targets may be tracked from any of the available film reels (camera sites) for the run. If the PDS computer cannot successfully locate enough targets (through computer-controlled tracking), the PDS operator generates another data tape in which (s)he manually digitizes (with the x-y crosshairs) the necessary targets.

The production runstream is designed to reprocess the new PDS data tape with a minimum of operator preparation. It is also equipped with the option to process redigitized data residing on the same PDS tape as the initial data (for the same film reel). The production runstream also creates UNIVAC data tapes directly after (a) conversion and (b) reformatting in order to make data available for further analysis, error correction, and reprocessing.

(3) The individual cameras (used in the experiments) are calibrated to provide mathematical camera constants vital in the computation of three-dimensional position (of the test subject). Examples of camera constants are the camera nodal point coordinates (x, y and z) and the lens distortion parameter. These values and other camera constants are generated by a camera-calibration program which uses as input converted data from a camera-calibration film reel. The photo data production runstream is designed to, with slight modification, execute the camera-calibration conversion program. This altered runstream is included as an element in the program file containing all photo data reduction programs.

(4) The modular organization of the steps (elements) in the photo data reduction process allows for efficient and expedient expansion of functions. New elements can be added without changing any existing elements. This feature affords the system a desirable flexibility. Standardized data formats and I/O access methods provide further flexibility. Possible additional functional elements are listed below:

(a) Error editing elements which correct improper data values in the data file.

(b) Packing elements which combine data files (and accompanying directories) from multiple tapes onto a single tape (with one all-inclusive directory).

(c) Plot elements which allow plot selection of individual photo variables residing on a converted photo data file.

(d) Special analysis elements which extract the data easily through the use of standardized data file formats and standardized I/O access methods.

Operational particulars of the runstream are as follows:

(1) Program File:

DATASYSTEMS*PHOTOSTREAM.

(2) Programming Languages:

(a) UNIVAC ASCII FORTRAN Level 10R1

(b) UNIVAC Executive System EXEC Level 38R5

(c) UNIVAC Text Editor Level 16R1

CONCLUSION

The photo data reduction and analysis system has been revised to successfully meet all proposed objectives:

1. All constituent programs are now compatible with announced changes in the UNIVAC operating system. These changes incorporate an ASCII FORTRAN programming environment.
2. A new graphics support capability is now utilized to:
 - a. Improve error detection procedures by providing additional plot presentations.
 - b. Produce output compatible with a variety of graphics devices (such as the Tektronix 4014® and the Hewlett Packard 7220®).
 - c. Reduce turn-around time of analysis by providing for immediate review of plots.
3. A stand-alone runstream providing for greater operator efficiency is now in effect. In addition, all component programs exist in a single program file.
4. Standardized data formats and I/O access methods, which in unison provide compatibility with stored archival data, are now incorporated into photo data reduction processing.
5. A modular organization of software is now incorporated in order to provide for greater operational flexibility.

PDS OUTPUT TAPE FORMAT

RECORD SEQUENCE

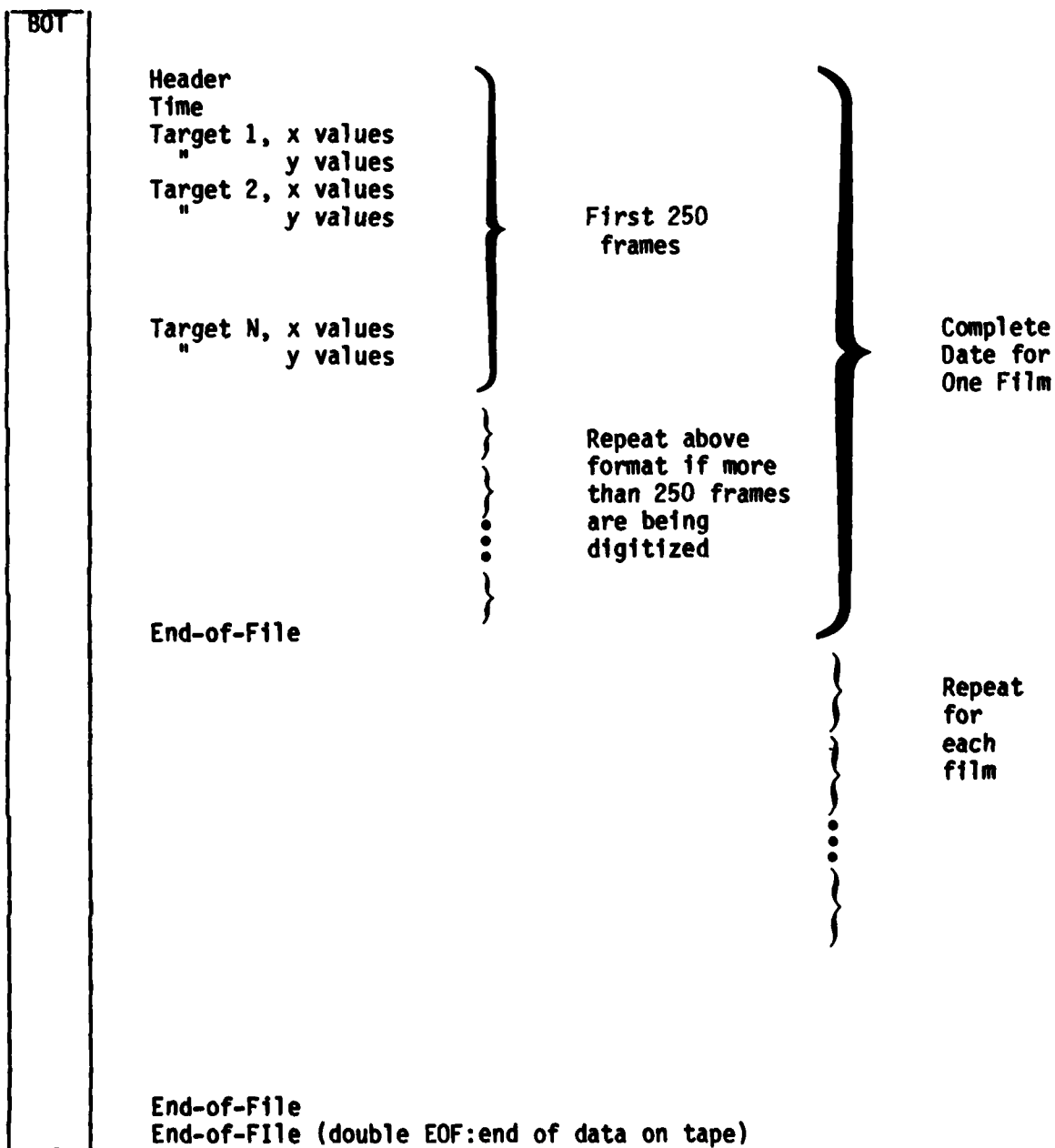


PHOTO HEADER RECORD FORMAT

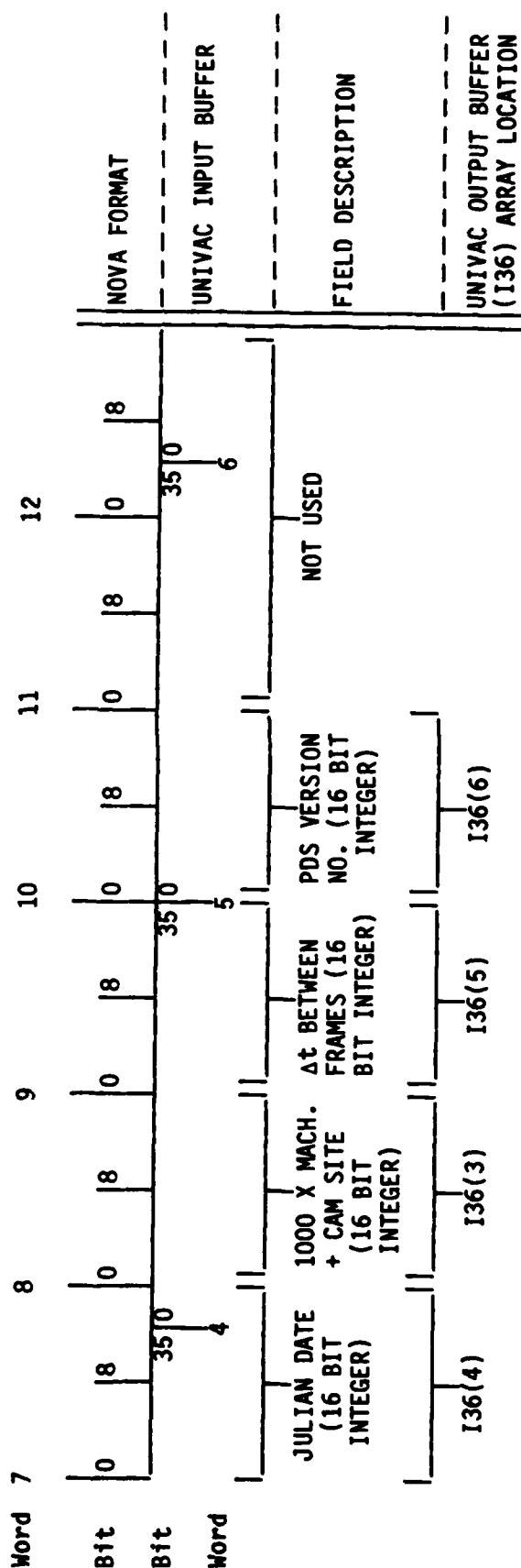
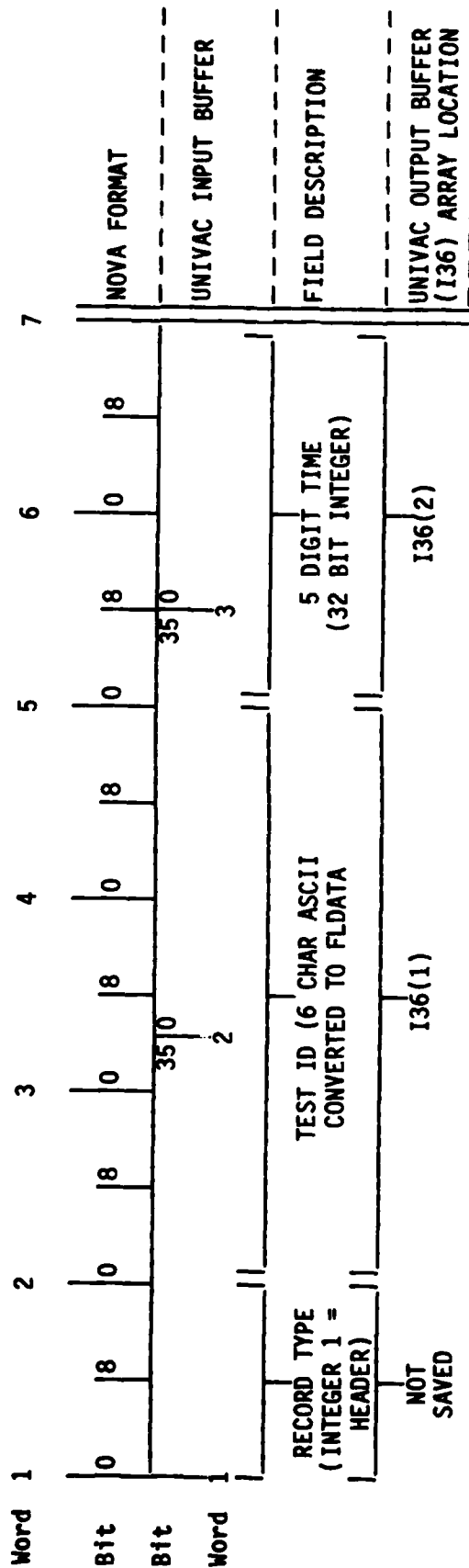


FIGURE 1b

28 Mar 79

PHOTO TIME DATA RECORD FORMAT

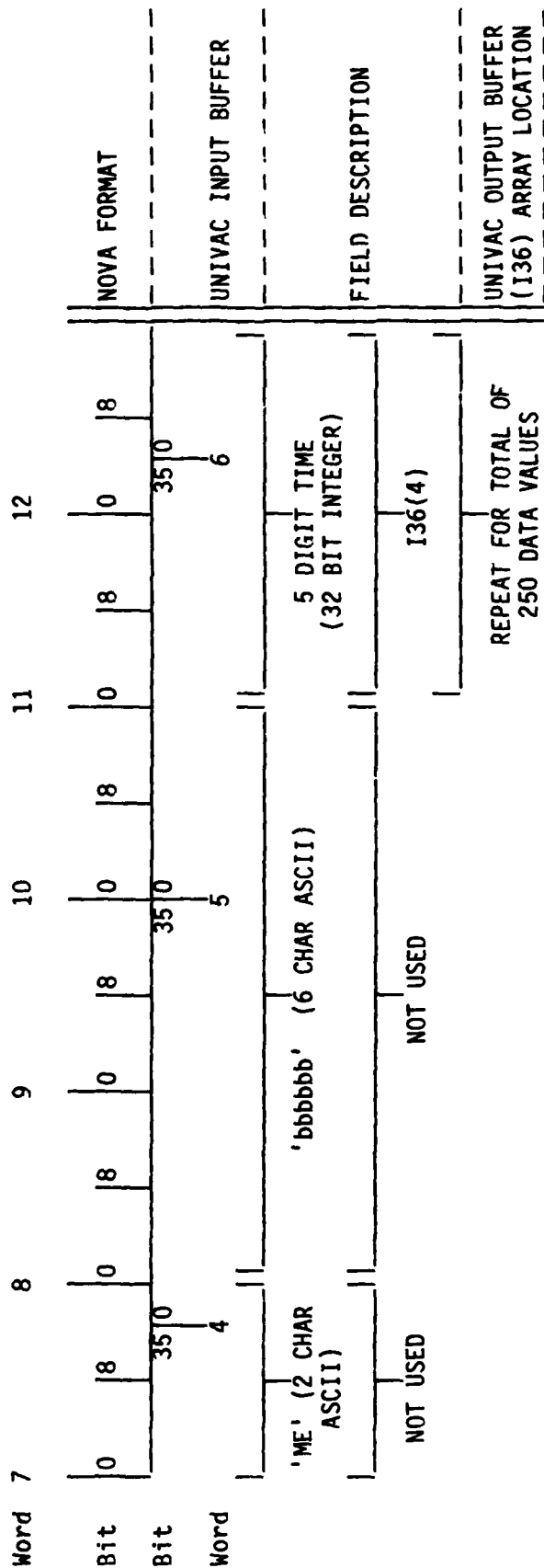
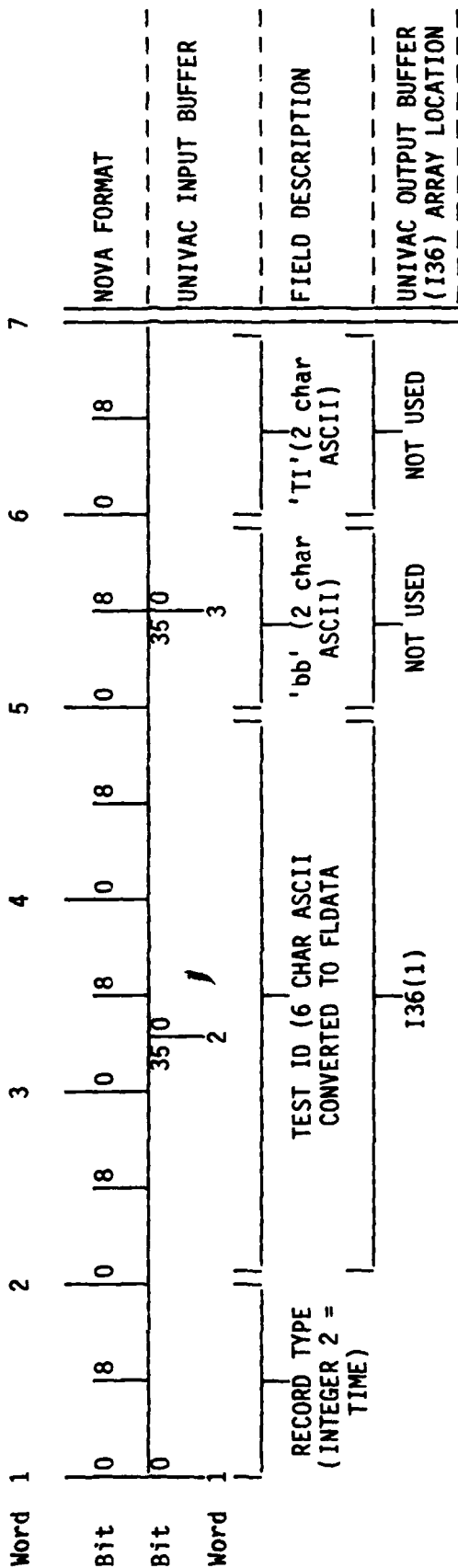


FIGURE 1c

28 Mar 79

PHOTO X (OR Y) DATA RECORD FORMAT

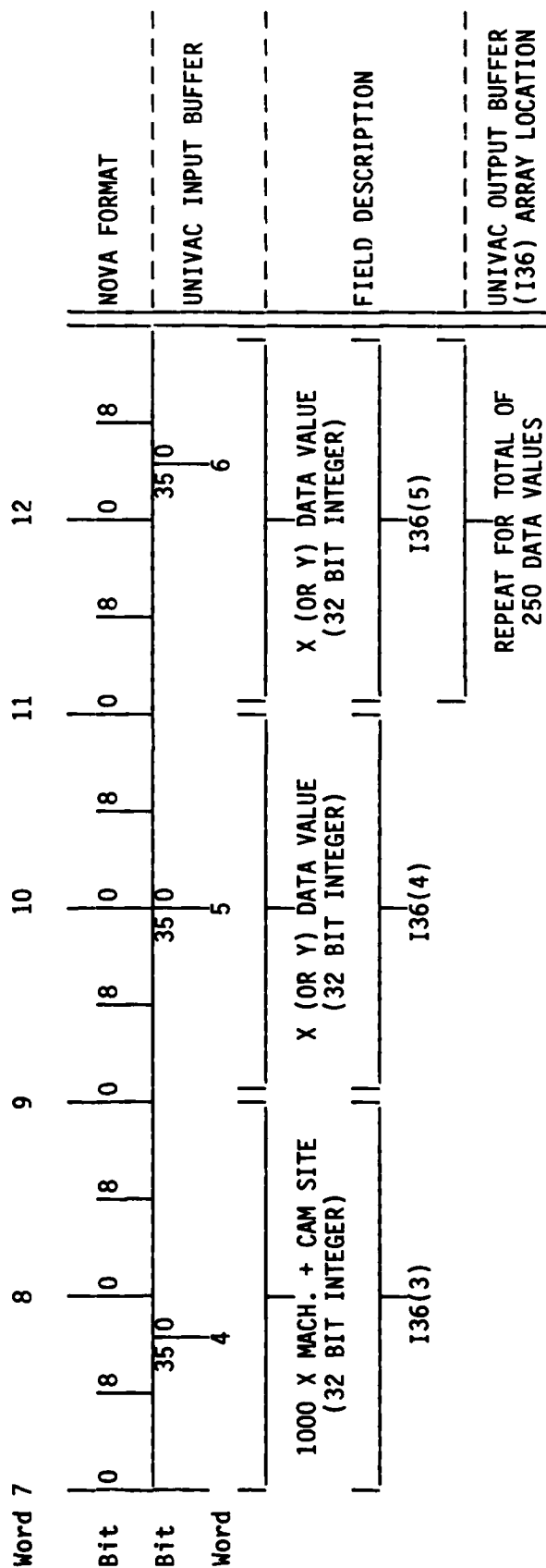
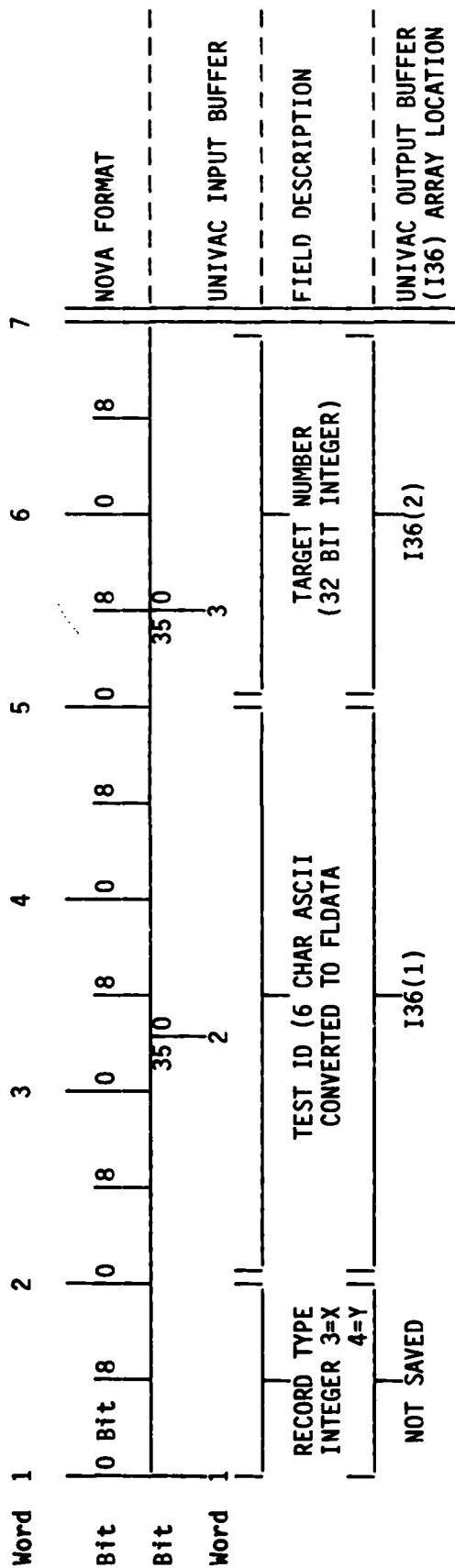


FIGURE 1d

28 Mar 79

HEADER FORMAT (PDS DATA)

<u>16 Bit Word</u>	<u>Contents</u>
1	Record type: 1 (16 bit integer)
2 }	Test ID (6 character ASCII)
3 }	
4 }	
5 }	Time of first frame in units of the least significant LED digit (32 bit integer)
6 }	
7	Julian date digitized: DDDY (16 bit integer)
8	1000 X machine # + camera site # (16 bit integer)
9	Δt : expected time between successive frames, in units of the least significant LED digit (16 bit integer)
10	Program version number (16 bit integer)
11	Not used
12	Not used

TIME RECORD FORMAT (PDS DATA)

<u>16 Bit Word</u>	<u>Contents</u>
1	Record type: 2 (16 bit integer)
2 } 3 } 4 }	Test ID (6 character ASCII)
5 } 6 } 7 } 8 } 9 }	'66 TIME 666666' (12 character ASCII)
10 }	
11	Time of frame 1 in units of the
12	least significant LED digit (32 bit integer)
.	
.	
509 }	Time of frame 250
510 }	
512	Not used
513	Not used (fixed length record)

X (OR Y) COORDINATE RECORD FORMAT (PDS DATA)

<u>16 Bit Word</u>	<u>Contents</u>
1	Record type: 3=x values (16 bit integer) 4=y values
2 }	
3 }	Test ID (6 character ASCII)
4 }	
5 }	Target # (32 bit integer)
6 }	
7 }	
8 }	Julian date on film: DDDY (16 bit integer)
9 }	x (or y) value for target in frame 1: displacement from a
10 }	fixed location in the filmplane in units of .00001 inches. Negative value indicates manual entry was made using the crosshairs. Zero indicates location of target was not defined for this frame. (32 bit integer)
.	
.	
509 }	x (or y) value for target in frame 250
510 }	
511	Not used
512	Not used (fixed length record)

COMPATIBLE PHOTO DATA

BOT	Location of random	}	5 words	}	
	Access table				
	Header	}	7 words		
	Time				
	Target 1, x values	}	253 words (each record)		
	" y values				
	Target 2, x values				
	" y values				
	Target N, x values	}			
	" y values				
End-of-file	}	Complete data for one run			
	}	Repeat above format if more than 250 words were digitized			
	}	Repeat for each run			
	}				
	}	End-of-file End-of-file End-of-file (Triple EOF: end of data) Random access table			

SITE*
ONE

May 5, 1982

25

RECORD FORMAT FOR LOCATION OF RANDOM
ACCESS TABLE FOR UNIVAC COMPATIBLE PHOTO DATA

<u>WORD</u>	<u>CONTENTS</u>
1	Sector address of random access table (integer)
2	Number of runs in this block of data (integer)
3	Site ID (integer)
4	Not used
5	Not used

May 5, 1982

Figure 2b

HEADER RECORD FORMAT FOR UNIVAC

COMPATIBLE PHOTO DATA

<u>FIELD</u>	<u>CONTENTS</u>
1	Run number (alpha - 6 characters) FLDATA
2	The word 'PHDATA' (alpha) FLDATA
3	Site ID (integer)
4	Time of first frame - 5 most significant digits (real)
5	Julian date: DDDY (alpha-6 char) Date digitized
6	Δt : expected time between frames (real)
7	Program version number (alpha-6 char)

TIME RECORD FORMAT FOR UNIVAC

COMPATIBLE PHOTO DATA

<u>FIELD</u>	<u>CONTENTS</u>
1	Run number (alpha - 6 characters) FLDATA
2	The word 'CTSbbb' (alpha) FLDATA*
3	Site ID (integer)
4	Time of first frame, 5 least significant digits (real)
.	
.	
.	
.	
.	
.	
.	
253	Time of 250th frame

* C is the camera site ID (1, 2, or 3)

DATA RECORD FORMAT FOR UNIVAC

COMPATIBLE PHOTO DATA

<u>WORD</u>	<u>CONTENTS</u>
1	Run number (alpha - 6 characters)
2	Displacement ID (alpha - 6 characters) (see Figure 4)
3	Site ID (integer)
4	X (or Y) value for target in frame 1: Displacement from a fixed location in meters. Negative value indicates manual entry was made using the crosshairs. A value of 999.0 indicates location of target was undefined for this frame. (REAL)
.	
.	
.	
.	
253	X (or Y) value for target in frame 250.

RANDOM ACCESS DIRECTORY FORMAT
FOR UNIVAC COMPATIBLE PHOTO DATA

<u>FIELD</u>	<u>CONTENTS</u>
0	Words following in this record (integer)
1	RN(1) Run number for first run (alpha - 6 characters) FLDATA
2	SH(1) Sector address of header for first run (integer)
3	NR(1) Number of physical records of 253 words required to make one logical record for each target for first run. This occurs when more than 250 frames are digitized for a run. Output is written in multiples of 253 word physical records (integer)
4	ST(1) Sector address of time record of first physical record for first run (integer)
.	.
.	.
.	.
.	.
NR(1)+3	ST(1,NR(1)) Sector address of time record of NR physical records for first run (integer)
For J number of runs where X is the number of words used by all previous runs	
1+X	RN(J) Run number for Jth run FLDATA
2+X	SH(J) Sector address of header for Jth run
3+X	NR(J) Number of physical records for Jth run
4+X	ST(J,1) Sector address of time record of 1st physical record for Jth run
.	.
.	.
.	.
.	.
NR(J)+X+3	ST(J,NR(J)) Sector address of time record of NR(J)th physical record for Jth run

See Figure 3 for example

EXAMPLE OF RANDOM ACCESS TABLE WITH
SINGLE PHYSICAL RECORD FOR EACH TARGET

<u>RUN NO.</u>	<u>SH</u>	<u>NR</u>	<u>ST(1)</u>
LX3036	1	1	2
LX3037	134	1	135
LX3038	267	1	268
LX3039	400	1	401
LX3041	533	1	534
LX3042	666	1	667
LX3045	799	1	800
LX3047	932	1	933
LX3048	1065	1	1066
LX3050	1198	1	1199
LX3051	1331	1	1332
LX3053	1504	1	1505
LX3049	1637	1	1638

Figure 3

DISPLACEMENT ID (EZFLOW FORMAT)

CDMTTb 6 char ID (Field data)

where

C: camera site ID (1, 2, or 3)

D: coordinate component (X or Y) of data

M: mount system

1: neck*

2: mouth

3: top of head

4: pelvis

5: other

9: sprocket hole**

TT: target ID (01, 02, 03, ...)

* The chair reference target has the special mount/target ID of 100

** The sprocket hole data has the special mount/target ID of 900

Figure 4

FILM FRAME PLOTS

LX4638 CAM 1 .00218 MOUTH PAGE 3

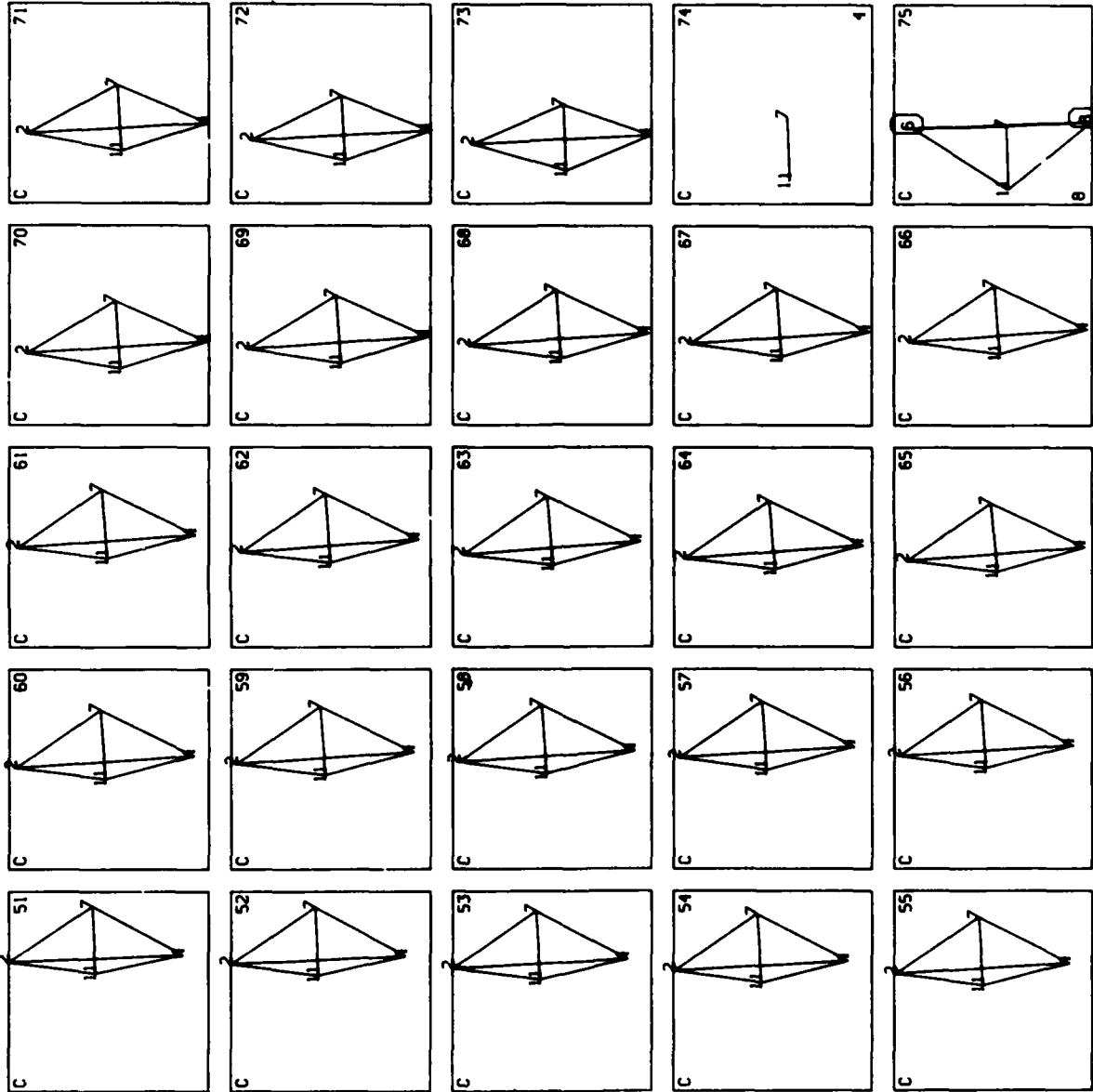


Figure 5a

LX4638 CAM 1 MOUTH 12 13 83

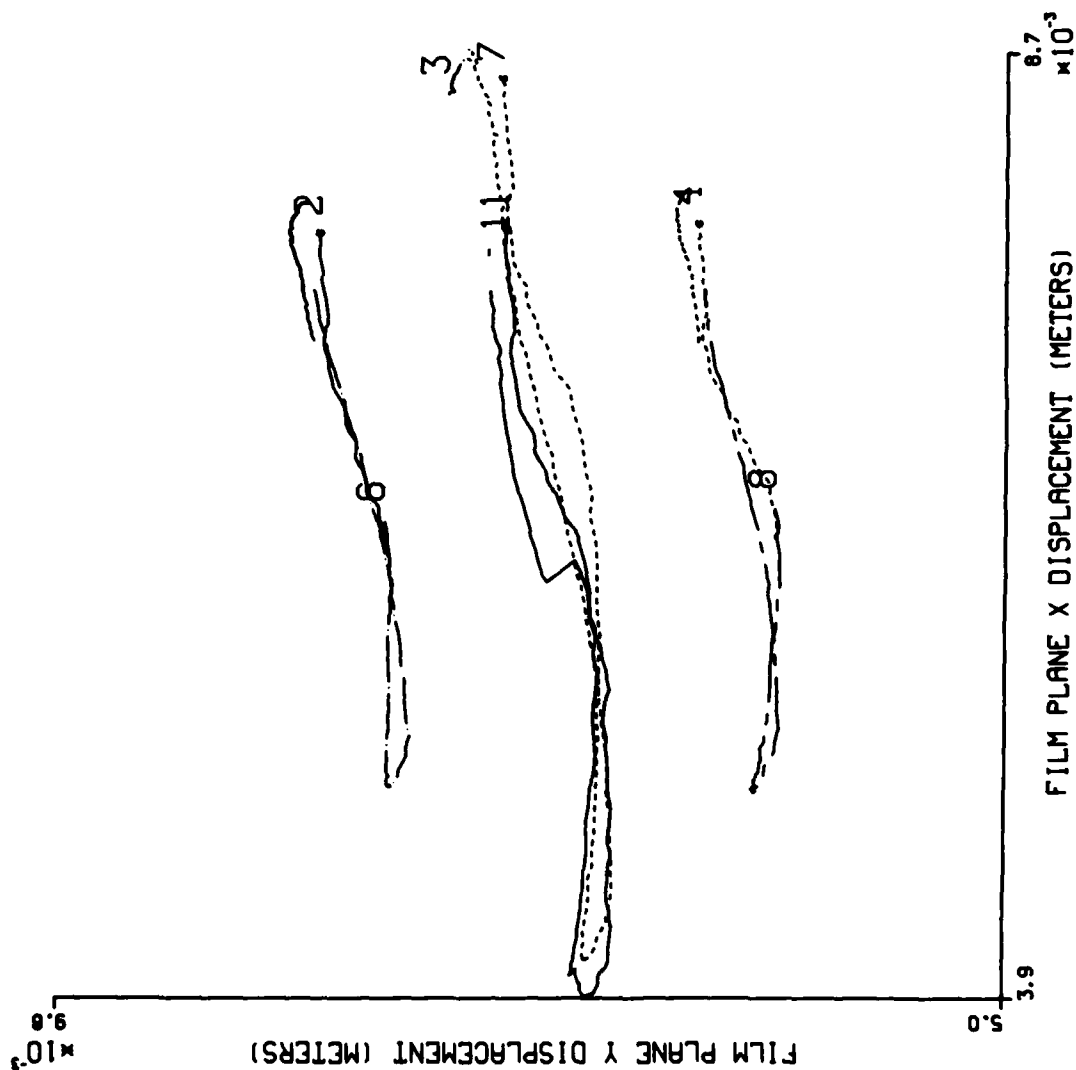
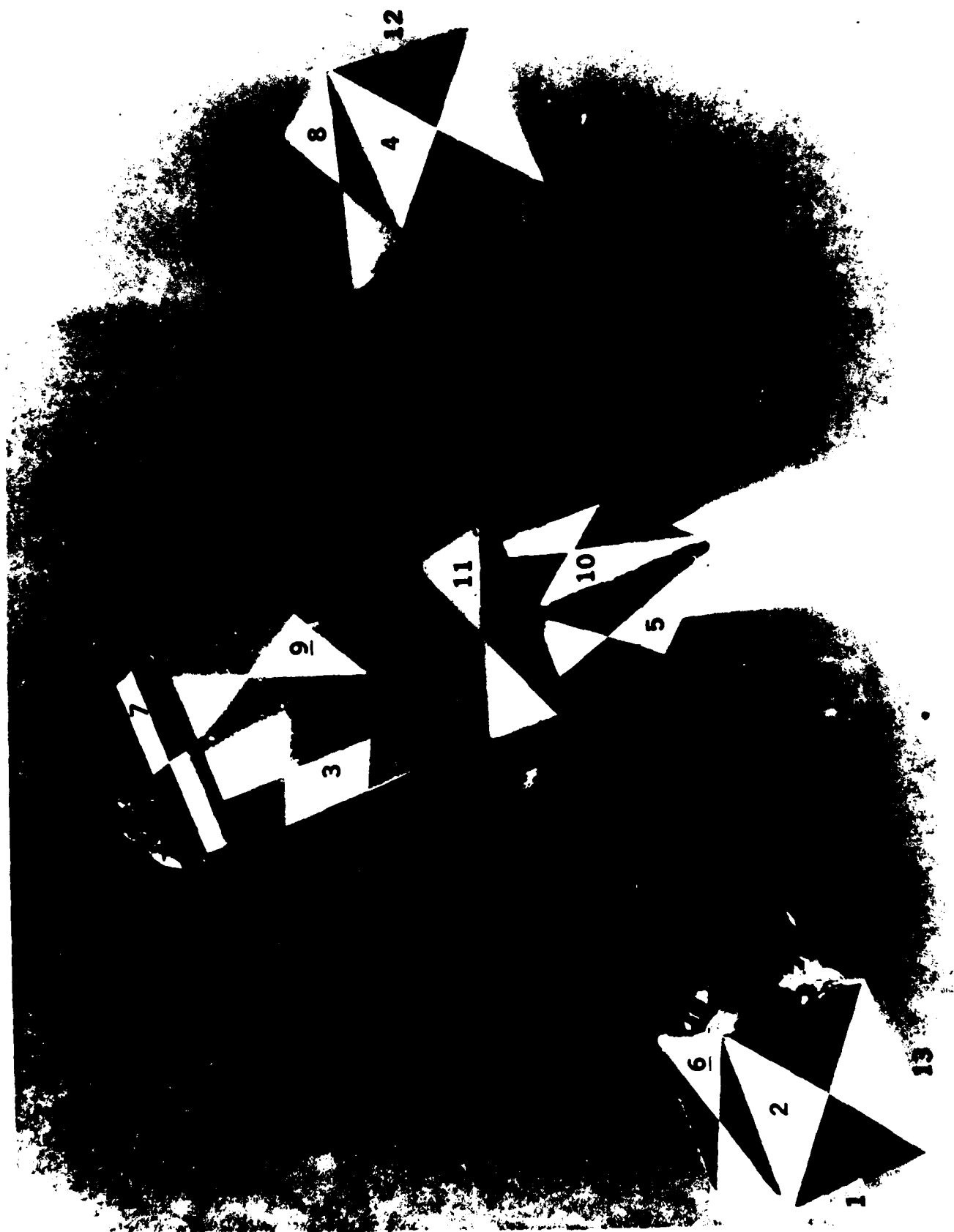


FIGURE 5b



ANATOMICAL (MOUTH) MOUNT TARGET LABELS

Figure 6

PHOTO DATA REDUCTION RUNSTREAM

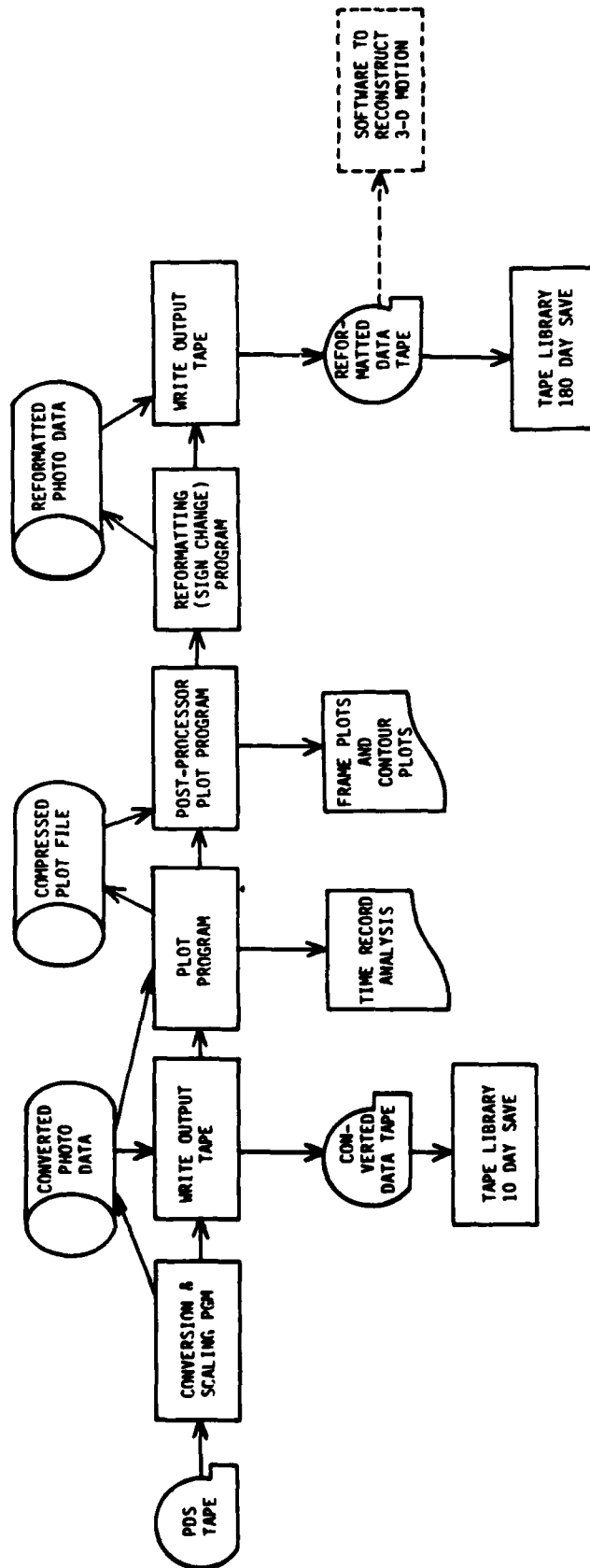


FIGURE 7

PHOTO DATA REDUCTION RUNSTREAM ELEMENT

```

@RUN
@PASSWD
@ASG,A DATASYSTEMS*PHOTOSTREAM.
@COPY DATASYSTEMS*PHOTOSTREAM.
@FREE DATASYSTEMS*PHOTOSTREAM.
@ED,U .XQTCVT
C /REELN/XXXXX/A
FIND@XQT
INPUT
( 0 0 0 0)
@EOF
@ADD,EP .XQTCVT
@ED,U .XQTPLT
FIND@XQT
INPUT
3 0 0 0 0 0 LX1111 LX9999
@EOF
@ADD,EP .XQTPLT
@ED,U .XQTSGN
C /XXXXXXX/11119999/A
FIND@XQT
INPUT
3 LX1111 LX9999
@EOF
@ADD,EP .XQTSGN
@FIN

```

```

ENTER EDIT PROCESSOR
WHERE XXXX=CURRENT PDS REEL NUMBER
LOCATE PLACE TO INSERT CORRECTIONS
INSERT CORRECTIONS (IF ANY) AS PER ELT .XQTCVT (APPENDIX 1a)
(INSET ZEROES IF NO CORRECTIONS REQUIRED)
EXIT INPUT MODE AND EDIT PROCESSOR
EXECUTE CONVERSION PROGRAM
ENTER EDIT PROCESSOR
LOCATE PLACE TO INSERT PLOT OPTIONS
INSERT PLOT OPTIONS AS PER ELT .XQTPLT (APPENDIX 2a)
REFER TO .XQTPLT FOR OPTION DEFINITIONS
EXIT INPUT MODE AND EDIT PROCESSOR
EXECUTE FRAME PLOT PROGRAM
ENTER EDIT PROCESSOR
WHERE 11119999=FILE NAME OF 180-DAY UNIVAC OUTPUT DATA TAPE
LOCATE PLACE TO INSERT INPUT OPTIONS
INSERT INPUT OPTIONS AS PER ELT .XQTSGN (APPENDIX 3a)
REFER TO .XQTSGN FOR OPTION DEFINITIONS
EXIT INPUT MODE AND EDIT PROCESSOR
EXECUTE REFORMATTING PROGRAM

```

FIGURE 8

REFERENCES

1. J. J. Lambert, "Development of ASCII FORTRAN I/O Program for Photo Data", (NBDL:72B:jbj, 5230, 1 June 1983), New Orleans, LA, 1983.
2. Integrated Software Systems Corporation, "DISSPLA User's Manual", San Diego, CA, 1981.
3. Sperry Corporation, "Sperry UNIVAC 1100 Executive System EXEC Level 38R5 Programmer Reference", Blue Bell, PA, 1982.
4. Sperry Corporation, "Sperry UNIVAC Series 1100 FORTRAN (ASCII) Level 10R1 Programmer Reference", Blue Bell, PA, 1982.
5. Sperry Corporation, "Sperry UNIVAC 1100 Series Text Editor (ED Processor) Level 16R1", Blue Bell, PA, 1980.

LAMBERT-TPFS(O).XQTCVT(26)

```

1  @ASG,TJ      1.,U9H,REELN . INPUT DATA FROM DIGITIZER
2  @ASG      9.,F///300
3  @ASG     10.,F///300
4  @ASG     11.,F///300
5  @
6  @XQT      MPO051/PHOTO C
7  12345678901234567890123456789012345678901234567890
8  @
9  @. ALL CONTROL STATEMENTS IMMEDIATELY AFTER @XQT STATEMENT
10 @. 1ST CONTROL STATEMENT
11 @. POS 5 - NRD = NO. OF RUNS FOR WHICH TO CHANGE RUN NOS. MAX OF 6
12 @. POS 10 - NRB = NO. OF RUNS TO BYPASS ON THIS TAPE. MAX OF 6
13 @. (SEE NOTE IN PGM MAIN)
14 @. POS 15 - NRC = NO. OF RUNS REQUIRING A CHANGE IN CAM NO. MAX OF 6
15 @
16 @. 2ND CONTROL STATEMENT (1 REQUIRED FOR EACH MAGNITUDE OF NRD)
17 @. POS 5-10 - LR = OLD RUN NUMBER
18 @. POS 15-20 - NR = NEW RUN NUMBER
19 @. POS 25 - IC = CAMERA NO. FOR WHICH RUN NO. IS TO BE CHANGED
20 @. POS 30 - IP = OCCURRENCE OF RUN NO. TO BE CHANGED, 1ST, 2ND, ETC.
21 @. MAY BE MORE THAN ONE OF ABOVE CONTROL STATEMENT
22 @
23 @. NEXT CONTROL STATEMENT (1 REQUIRED FOR EACH MAGNITUDE OF NRB)
24 @. POS 5-10 - IRN = RUN NUMBER TO BYPASS
25 @. POS 15 - ICN = CAMERA NO. FOR WHICH TO BYPASS RUN
26 @. MAY BE MORE THAN ONE OF ABOVE CONTROL STATEMENT
27 @
28 @. NEXT CONTROL STATEMENT (1 REQUIRED FOR EACH MAGNITUDE OF NRC)
29 @. POS 5-10 - LRN = RUN NUMBER
30 @. POS 15 - LCN = OLD CAMERA NUMBER
31 @. POS 20 - NCN = NEW CAMERA NUMBER
32 @. POS 25 - NOC = OCCURRENCE OF CAMERA TO CHANGE, 1ST, 2ND, ETC.
33 @. MAY BE MORE THAN ONE OF ABOVE CONTROL STATEMENT
34 @
35 @FREE      1.
36 @TEST     TNE/O/S6
37 @JUMP     OUT
38 @ASG,TF    A.,U9S,SAVE01 . PREPLOTPHOTODATA
39 @COPY,GM   9.,A.
40 @TEST2:
41 @TEST     TG/1/S6
42 @JUMP     TEST3
43 @COPY,GM   10.,A.
44 @TEST3:
45 @TEST     TG/2/S6
46 @JUMP     OUT
47 @COPY,GM   11.,A.
48 @OUT:
49 @FREE     A.
50 @. EXITED FROM .XQTCVT

```

@PRT,S .MAIN

```

LAMBERT+TPF$(O).MAIN(37)
1      C   THIS PROGRAM WILL READ DIGITIZED PHOTODATA IN PDS OUTPUT
2      C   FORMAT AND CONVERT TO UNIVAC 1182 FIELDATA FORMAT.  A
3      C   SEPARATE FILE IS GENERATED FOR EACH CAMERA SITE ID.
4      C
5      C   NRD = NUMBER OF RUNS REQUIRING A CHANGE IN THE RUN NUMBER.
6      C   THE RUN NUMBER MAY OR MAY NOT BE DUPLICATED.
7      C   LR = OLD RUN NUMBER
8      C   NR = NEW RUN NUMBER
9      C   IC = CAMERA NUMBER FOR WHICH A RUN NUMBER IS TO BE CORRECTED
10     C   IP = OCCURRENCE OF RUN NUMBER TO BE CHANGED, 1ST, 2ND, ETC.
11     C
12     C
13     C   NRB = NUMBER OF RUNS TO BYPASS ON THIS TAPE
14     C   IRN = RUN NUMBER TO SKIP
15     C   ICN = CAMERA NUMBER FOR WHICH TO SKIP RUN
16     C
17     C   IT IS ASSUMED THAT WHEN A RUN IS TO BE BYPASSED, IT HAS
18     C   BEEN PREVIOUSLY DIGITIZED AND PLACED IN THE FILE, AND
19     C   LATER RE-DIGITIZED AND PLACED IN THE FILE AFTER A GROSS
20     C   ERROR WAS DISCOVERED.  THE FIRST OCCURRENCE OF THE RUN IS
21     C   ALWAYS BYPASSED.
22     C
23     C
24     C   NRC = NUMBER OF RUNS REQUIRING A CHANGE IN CAMERA NUMBER
25     C   LRN = RUN NUMBER
26     C   LCN = OLD CAMERA NUMBER
27     C   NCN = NEW CAMERA NUMBER
28     C   NOC = OCCURRENCE OF CAMERA NUMBER TO BE CHANGED, 1ST, 2ND, ETC
29     C
30     C
31     C   DIMENSION LCN(6), NCN(6), NOC(6), ITRY(6), ICCD(6)
32     C   DIMENSION IC(6), IP(6), ICN(6), IPAS(6)
33     C   DIMENSION ITABLE(5), JTABLE(4000), ICAM(3)
34     C   COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
35     C   COMMON/I36/IOUT(3),DATA(250),IDL,IUNIT,ITARG
36     C   CHARACTER*6 RUNID,VARID,LR(6),NR(6),IRN(6),FD2ASC,
37     C   1LRN(6),JDTDIG,IPGVER,RUN(100,3)
38     C
39     C
40     C   DATA IOUT/9,10,11/
41     68   FORMAT(1X,I4)
42     C   IERR=0
43     C   IUNIT=1
44     C
45     C   READ CORRECTION INFORMATION
46     C
47     C   READ (5,230) NRD,NRB,NRC
48     C   PRINT 230, NRD,NRB,NRC
49     C
50     C   IF (NRD.EQ.0) GO TO 20
51     C   DO 10 J=1,NRD
52     C     READ (5,240) LR(J),NR(J),IC(J),IP(J)
53     C     PRINT 240, LR(J),NR(J),IC(J),IP(J)
54     C   CONTINUE
55     C
56     20   IF (NRB.EQ.0) GO TO 40
57     C   DO 30 J=1,NRB
58     C     READ (5,250) IRN(J),ICN(J)
59     C     PRINT 250, IRN(J),ICN(J)

```



```

60      30      CONTINUE
61      C
62      40      IF (NRC.EQ.0) GO TO 80
63      DO 50 J=1,NRC
64          READ (5,250) LRN(J),LCN(J),NCN(J),NOC(J)
65          PRINT 250, LRN(J),LCN(J),NCN(J),NOC(J)
66      50      CONTINUE
67      C
68      60      CONTINUE
69      C
70      NC=0
71      C
72      70      CONTINUE
73      C
74      C      READ RECORD FROM PDS TAPE
75      C
76      CALL CONV(ITYPE,IST)
77      IF (IST.NE.4) GO TO 210
78      C
79      IF (ITYPE.NE.1) GO TO 70
80      IF (NRD.EQ.0) GO TO 100
81      DO 90 J=1,NRD
82          IF (RUNID.EQ.LRN(J).AND.ISITID.EQ.IC(J)) ITRY(J)=ITRY(J)+1
83          IF (RUNID.EQ.LRN(J).AND.ISITID.EQ.IC(J).AND.ITRY(J).EQ.IP(J))
84      1  RUNID=NR(J)
85      90      CONTINUE
86      C
87      100     IF (NRB.EQ.0) GO TO 120
88      DO 110 J=1,NRB
89          IF (RUNID.EQ.IRN(J).AND.ISITID.EQ.ICN(J).AND.IPAS(J).EQ.0)
90      1  GO TO 190
91      110     CONTINUE
92      C
93      C      PROCESS RUNS CONTAINING INCORRECT CAMERA NUMBERS
94      C
95      120     IF (NRC.EQ.0) GO TO 140
96      DO 130 J=1,NRC
97          IF (RUNID.EQ.LRN(J).AND.ISITID.EQ.LCN(J))
98      1  ICCD(J)=ICCD(J)+1
99          IF (RUNID.EQ.LRN(J).AND.ISITID.EQ.LCN(J).AND.NOC(J).EQ.
100      1  ICCD(J)) ISITID=NCN(J)
101      130     CONTINUE
102      C
103      140     CONTINUE
104      C
105      ISW=ISITID
106      IF (ISW.LE.0.OR.ISW.GT.3) GO TO 150
107      C
108      C      DETERMINE WHICH CAMERA NUMBER TO PROCESS
109      C
110      GO TO (160,170,180), ISW
111      C
112      150     CONTINUE
113      PRINT 270, RUNID,ISITID,ISW
114      STOP 'CAM NO. ERROR'
115      C
116      160     CONTINUE
117      C
118      C      CONVERT DATA FOR CAMERA 1
119      C

```

```

120      CALL CNVT1(ITYPE,IST)
121      IF (NC.LT.2) NC=1
122      GO TO 70
123      C
124      170  CONTINUE
125      C
126      C      CONVERT DATA FOR CAMERA 2
127      C
128      CALL CNVT1(ITYPE,IST)
129      IF (NC.LT.3) NC=2
130      GO TO 70
131      C
132      180  CONTINUE
133      C
134      C      CONVERT DATA FOR CAMERA 3
135      C
136      CALL CNVT1(ITYPE,IST)
137      NC=3
138      GO TO 70
139      C
140      190  CONTINUE
141      IPAS(J)=1
142      C
143      C      BYPASS CURRENT RUN
144      C
145      200  CALL CNVT(ITYPE,IST)
146      IF (IST.EQ.4) GO TO 200
147      GO TO 70
148      C
149      210  CONTINUE
150      C
151      C      WRITE TRIPLE END-OF -FILE AT END OF DATA
152      C
153      DO 220,I=1,NC
154      DO 225,J=1,3
155      CALL EOFPUT(IOUT(I),IER)
156      IF(IER.NE.0)THEN
157      PRINT 13,IER
158      STOP 'ABORT IN MAIN'
159      END IF
160      225  CONTINUE
161      220  CONTINUE
162      C
163      C      SET CONDITION WORD TO NUMBER OF CAMERAS
164      C
165      CALL FCOND(ICOND)
166      BITS(ICOND,25,12)=BITS(NC,25,12)
167      CALL FSETC(ICOND)
168      C
169      C      PRINT DIRECT ACCESS TABLES
170      C
171      PRINT 251
172      MAX=4000
173      DO 300,I=1,NC
174      CALL GETDIR(IOUT(I),MAX,0,ITABLE,IER)
175      IF(IER.NE.0)STOP 'DIRECTORY ERR'
176      ICAM(I)=ITABLE(3)
177      NRUNS=ITABLE(2)
178      ISECT=ITABLE(1)
179      PRINT 211,ICAM(I)

```

```

180      CALL GETDIR(IOUT(I),MAX,ISECT,JTABLE,IER)
181      IF(IER.NE.O)STOP 'DIRECTORY ERR'
182      IPRINT=2
183      DO 296,J=1,NRUNS
184      RUN(NRUNS,ICAM(I))=FD2ASC(JTABLE(IPRINT),6)
185      PRINT 221,RUN(NRUNS,ICAM(I)),(JTABLE(K),K=IPRINT+1,IPRINT+1+
186      1JTABLE(IPRINT+2)+1)
187      IPRINT=IPRINT+3+JTABLE(IPRINT+2)
188      296  CONTINUE
189      PRINT 231,(ITABLE(K),K=1,5)
190      300  CONTINUE
191      C
192      STOP
193      C
194      13  FORMAT(1X,'STATUS ERROR = ',03)
195      230  FORMAT (3I5)
196      240  FORMAT (2(4X,A6),2I5)
197      250  FORMAT (4X,A6,3I5)
198      260  FORMAT (' READ ERROR, STATUS = ',I3)
199      270  FORMAT (' CAMERA NUMBER ERROR, RUN = ',A6,2I10)
200      251  FORMAT('1')
201      211  FORMAT(' DIRECT ACCESS TABLE FOR CAMERA ',I2)
202      231  FORMAT(1X,5(16,4X)//)
203      221  FORMAT(1X,A6,2X,13I8)
204      C
205      END

```

•PRT,S .CONVT

```

LAMBERT+TPF$(0).CNVT1(7)
1      SUBROUTINE CNVT1(ITYPE,IST)
2      C
3      C      FUNCTION:
4      C      CONTROL (1) READ OF DIGITIZED PHOTO DATA FROM PDS TAPE
5      C      (2) CONVERSION TO UNIVAC COMPATIBLE FORMAT AND (3) SCALING
6      C      OF DATA
7      C
8      C      ARGUMENT DEFINITIONS:
9      C      ITYPE-RETURNED PDS RECORD TYPE WHERE
10     C      1=HEADER RECORD
11     C      2=TIME RECORD
12     C      3=X DATA RECORD
13     C      4=Y DATA RECORD
14     C      IST-RETURNED ERROR STATUS WORD FROM ADTIO GENERAL
15     C      PURPOSE I/O ROUTINE
16     C
17     COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
18     COMMON/I36/IOUT(3),DAT(250),IDL,IUNIT,ITARG
19     CHARACTER*6 VAR1C1(28), VAR2C1(28), VAR3C1(28), VARCC1(28)
20     CHARACTER*6 VAR1C2(28), VAR2C2(28), VAR3C2(28), VARCC2(28)
21     CHARACTER*6 VAR3C3(28)
22     CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,B(28),TARNUM*3,CAMNO*1
23     1,OLDRUN
24     C
25     C      DEFINE MOUNT/TARGET ID'S
26     C
27     DATA VAR1C1 /'1X100','1Y100','1X102','1Y102',
28     1'1X103','1Y103','1X101','1Y101','1X201','1Y201',
29     2'1X204','1Y204','1X203','1Y203','1X308','1Y308',12* '/'
30     DATA VAR2C1 /'1X100','1Y100','1X103','1Y103','1X105',
31     1'1Y105','1X104','1Y104','1X101','1Y101','1X201','1Y201','1X204',
32     2'1Y204','1X205','1Y205','1X203','1Y203','1X310','1Y310',8* '/'
33     DATA VAR3C1 /'1X100','1Y100','1X212','1Y212','1X204',
34     1'1Y204','1X208','1Y208','1X210','1Y210','1X202','1Y202','1X211',
35     2'1Y211','1X209','1Y209','1X207','1Y207','1X010','1Y010',8* '/'
36     DATA VARCC1 /'1X100','1Y100','1X002','1Y002','1X003',
37     1'1Y003','1X004','1Y004','1X005','1Y005','1X201','1Y201','1X204',
38     2'1Y204','1X205','1Y205','1X203','1Y203','1X310','1Y310',8* '/'
39     DATA VAR1C2 /'2X100','2Y100','2X203','2Y203',
40     1'2X204','2Y204','2X202','2Y202',20* '/'
41     DATA VAR2C2 /'2X100','2Y100','2X205','2Y205','2X204',
42     1'2Y204','2X202','2Y202','2X203','2Y203',18* '/'
43     DATA VAR3C2 /'2X100','2Y100','2X103','2Y103',
44     1'2X105','2Y105','2X101','2Y101','2X102',
45     2'2Y102','2X201','2Y201','2X202','2Y202','2X205','2Y205','2X204',
46     3'2Y204','2X203','2Y203','2X011','2Y011','2X012','2Y012',4* '/'
47     DATA VARCC2 /'2X100','2Y100','2X103','2Y103','2X105','2Y105',
48     1'2X101','2Y101','2X102','2Y102','2X201','2Y201','2X202',
49     2'2Y202','2X205','2Y205','2X204','2Y204','2X203','2Y203',8* '/'
50     DATA VAR3C3 /'3X100','3Y100','3X106','3Y106','3X111','3Y111',
51     1'3X102','3Y102','3X101','3Y101','3X103','3Y103','3X107','3Y107',
52     2'3X108','3Y108','3X105','3Y105','3X104','3Y104','3X109','3Y109',
53     3'3X110','3Y110',4* '/'
54     C
55     OLDRUN=' '
56     C
57     C
58     C      CONVERTS AND SCALES PHOTO DATA
59     C

```

```

60      DO 90 KP=1,1000
61      DO 10 J=1,250
62      10  DAT(J)=0.0
63      IF (ITYPE.NE.1) GO TO 20
64      C
65      C
66      C  CHECK FOR MORE THAN 250 FRAMES OF DATA.
67      C
68      C
69      IF (RUNID.EQ.OLDRUN) GO TO 30
70      OLDRUN=RUNID
71      VARID='PHDATA'
72      C
73      C  WRITE HEADER
74      C
75      CALL PUTRCD(IOUT(ISITID),RUNID,VARID,ISITID,IDL,DAT,IER)
76      IF (IER.NE.0) GO TO 112
77      PRINT 190,RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER,ITYPE
78      C
79      C  READ NEXT PDS RECORD
80      C
81      20  CALL CONVT(ITYPE,IST)
82      IF (IST.NE.4) GO TO 100
83      IF (ITYPE.EQ.1) GO TO 90
84      GO TO 40
85      C
86      C
87      C  HANDLES EOF'S FOR MORE THAN 250 FRAMES
88      C
89      30  CALL EOFPUT(IOUT(ISITID),IER)
90      IF (IER.NE.0) GO TO 112
91      CALL CONVT(ITYPE,IST)
92      IF (IST.NE.4) GO TO 100
93      C
94      C
95      C  DETERMINE VARIABLE NAMES FOR A GIVEN CAMERA.
96      C
97      C
98      40  C
99      C  CONTINUE
100     CHANGE DATA TO METERS & SET ZERO VALUES IN NON-TIME RECORDS EQUAL
101     VARID=' '
102     ENCODE (1,160,CAMNO) ISITID
103     SUBSTR(VARID,1,1)=CAMNO
104     IF (ITYPE.EQ.2) SUBSTR(VARID,2,2)='TS'
105     IF (ITYPE.EQ.2) GO TO 70
106     IF (BITS(ITARG,5,16).NE.0) GO TO 60
107     IF (RUNID.LT.'LX0253') NDT=1
108     IF (RUNID.LT.'LX1410') NDT=2
109     IF (RUNID.GE.'LX1410') NDT=3
110     IF ((RUNID.EQ.'LX1234').OR.(RUNID.EQ.'LX1235').OR.(RUNID.EQ.
111     1'LX1236').OR.(RUNID.EQ.'LX1237')) NDT=4
112     DO 50 J=1,28
113     IF ((ISITID.EQ.1).AND.(NDT.EQ.1)) B(J)=VAR1C1(J)
114     IF ((ISITID.EQ.1).AND.(NDT.EQ.2)) B(J)=VAR2C1(J)
115     IF ((ISITID.EQ.1).AND.(NDT.EQ.3)) B(J)=VAR3C1(J)
116     IF ((ISITID.EQ.1).AND.(NDT.EQ.4)) B(J)=VARCC1(J)
117     IF ((ISITID.EQ.2).AND.(NDT.EQ.1)) B(J)=VAR1C2(J)
118     IF ((ISITID.EQ.2).AND.(NDT.EQ.2)) B(J)=VAR2C2(J)
119     IF ((ISITID.EQ.2).AND.(NDT.EQ.3)) B(J)=VAR3C2(J)
120     IF ((ISITID.EQ.2).AND.(NDT.EQ.4)) B(J)=VARCC2(J)

```

```

120      IF ((ISITID.EQ.3).AND.(NDT.EQ.3)) B(J)=VAR3C3(J)
121      50  CONTINUE
122      ITT=2*ITARG
123      IF (ITYPE.EQ.3) VARID=B(ITT-1)
124      IF (ITYPE.EQ.4) VARID=B(ITT)
125      GO TO 70
126      60  CONTINUE
127      IF(ITYPE.EQ.3)SUBSTR(VARID,2,1)='X'
128      IF(ITYPE.EQ.4)SUBSTR(VARID,2,1)='Y'
129      NTC=BITS(ITARG,5,16)
130      ENCODE (3,170,TARNUM) NTC
131      SUBSTR(VARID,3,3)=TARNUM
132      70  CONTINUE
133      DO 80 J=1,250
134          IF (ITYPE.EQ.2) DAT(J)=DAT(J)/10000.0
135          IF (ITYPE.GT.2) DAT(J)=(DAT(J)/100000.0)*.0254
136          IF ((ITYPE.GT.2).AND.(DAT(J).EQ.0.0)) DAT(J)=999.0
137      80  CONTINUE
138          IF ((RUNID.GE.'LX0253').AND.(VARID.EQ.' ')) GO TO 90
139      C
140      C  CALL SPROCKET HOLE SUBROUTINE IF 900 OR 100 RECORD
141      C
142          IF (SUBSTR(VARID,2,5).EQ.'X900 ') CALL SPH9X (DAT)
143          IF (SUBSTR(VARID,2,5).EQ.'Y900 ') CALL SPH9Y (DAT)
144          IF (SUBSTR(VARID,2,5).EQ.'X100 ') CALL SPH1X (DAT)
145          IF (SUBSTR(VARID,2,5).EQ.'Y100 ') CALL SPH1Y (DAT)
146      C
147      C
148      C  WRITE TIME OR DISPLACEMENT RECORD
149      C
150          CALL PUTRCD(IOUT(ISITID),RUNID,VARID,ISITID,IDL,DAT,IER)
151          IF(IER.NE.0)GO TO 112
152          PRINT 200,RUNID,VARID,ISITID,ITYPE,(DAT(J1),J1=1,10)
153      90  CONTINUE
154      100 CONTINUE
155      C
156      C  WRITE DOUBLE END-OF-FILE AT END OF RUN (END OF
157      C  DATA FOR 1 FILM)
158      C
159          CALL EOFPUT(IOUT(ISITID),IER)
160          IF(IER.NE.0)GO TO 112
161          CALL EOFPUT(IOUT(ISITID),IER)
162          IF(IER.NE.0)GO TO 112
163      C
164      C  CALCULATE SPROCKET HOLE AVERAGES
165      C
166          CALL SPHAVG (SPHX,SPHY)
167          PRINT 180, SPHX,SPHY
168      C
169          RETURN
170      C
171      C  OUTPUT ABORT ROUTINE
172      C
173      112 PRINT 12,IER
174      12  FORMAT(1X,'STATUS ERROR = ',03)
175          STOP 'ABORT'
176      C
177      160 FORMAT (I1)
178      170 FORMAT (I3)
179      180 FORMAT (/' SPROCKET HOLE AVERAGES, X = ',F10.6,' Y = ',F10.6)

```

```

180      190  FORMAT ( '1', 1X, 2(A6,4X), 16,4X, F10.3,4X, A6,4X, F6.0,4X, A6,4X, 16)
181      200  FORMAT ( 1X, 2(A6,4X), 16,4X, 16,4X/ 1X, 10(F11.8,2X))
182      C
183      END

```

•PRT, S .SPH AVG

```

LAMBERT*TPF$(0).CONVT(34)
1      SUBROUTINE CONVT(ITYPE,IST)
2      C
3      C      FUNCTION:
4      C      (1) READ PDS PHOTO DATA FROM TAPE AND (2) CONVERT
5      C      DIGITIZED PHOTO DATA TO UHNIVAC COMPATIBLE FORMAT
6      C
7      C      ARGUMENT DEFINITIONS:
8      C      ITYPE-RETURNED PDS RECORD TYPE WHERE
9      C      1=HEADER RECORD
10     C      2=TIME RECORD
11     C      3=X DATA RECORD
12     C      4=Y DATA RECORD
13     C      IST-RETURNED ERROR STATUS WORD FROM ADTIO GENERAL
14     C      PURPOSE I/O ROUTINE
15     C
16     C      DIMENSION I32(300)
17     C      COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
18     C      COMMON/I36/IDUT(3),DAT(250),IDL,IUNIT,ITARG
19     C      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER
20     C
21     C      DATA FOR EXTENDING THE SIGN BIT.
22     C
23     C      DATA NZ /07400000000000/ ,SIGN /00200000000000/
24     C
25     C      NW=255
26     C      CALL ATRD(IUNIT,I32,NW,IST)
27     C      IF (IST.EQ.1) RETURN
28     C
29     C      ZERO OUT RECEIVING ARRAY.
30     C
31     C      DO 10 I=1,250
32     C          DAT(I)=0.
33     C      CONTINUE
34     C
35     C      CONVERT RUN NUMBER FROM NOVA ASCII TO UNIVAC ASCII
36     C
37     C      CALL ASCBT9(I32,1,17,6,RUNID)
38     C
39     C      ITYPE=BITS(I32(1),1,16)
40     C      IF (ITYPE.GT.1) GO TO 60
41     C
42     C      EXTRACT HEADER RECORD AND CONVERT TO UNIVAC WORDS
43     C
44     C      BITS(itime,5,8)=BITS(I32(2),29,8)
45     C      BITS(itime,13,24)=BITS(I32(3),1,24)
46     C      F1TIME=FLOAT(itime)
47     C      BITS(ISITID,21,16)=BITS(I32(4),5,16)
48     C      BITS(IDATE,21,12)=BITS(I32(3),25,12)
49     C      BITS(IDATE,33,4)=BITS(I32(4),1,4)
50     C      ENCODE(6,151,JDTDIG)IDATE
51     C      BITS(IDELTA,33,4)=BITS(I32(5),1,4)
52     C      DELTA=FLOAT(IDELTA)
53     C      BITS(IVER,25,12)=BITS(I32(5),5,12)
54     C      ENCODE(6,151,IPGVER)IVER
55     C      151  FORMAT(I6)
56     C      IF(IDATE.EQ.0)JDTDIG=' '
57     C      IF(IVER.EQ.0)IPGVER=' '
58     C
59     C      RETURN

```



```

60      60      IF (ITYPE.GT.2) GO TO 70
61      GO TO 80
62      C
63      C      EXTRACT TARGET NUMBERS FROM INPUT BUFFER
64      C
65      70      BITS(ITARG,5,8)=BITS(I32(2),29,8)
66      BITS(ITARG,13,24)=BITS(I32(3),1,24)
67      BITS(ISITID,5,12)=BITS(I32(3),25,12)
68      BITS(ISITID,17,20)=BITS(I32(4),1,20)
69      C
70      C      EXTRACT DATA VALUES FOR DATA RECORD (OR TIME RECORD)
71      C
72      IW=4
73      IB=21
74      GO TO 90
75      80      IW=5
76      IB=17
77      90      DO 110 I=1,250
78          ITEMP=0
79          IS=5
80          IL=37-IB
81          IF (IL.GT.32) IL=32
82          BITS(ITEMP,IS,IL)=BITS(I32(IW),IB,IL)
83          IF (IL.EQ.32) GO TO 100
84          IW=IW+1
85          IB=1
86          IS=4+IL+1
87          IL=37-IS
88          BITS(ITEMP,IS,IL)=BITS(I32(IW),IB,IL)
89      100      IB=IB+IL
90          IF (IB.GT.36) IW=IW+1
91      C
92      C      EXTEND THE SIGN BIT FOR NEGATIVE NUMBERS AND CHANGE
93      C      2'S COMPLEMENT TO 1'S COMPLEMENT
94      C
95          IF (AND(SIGN,ITEMP).NE.0) ITEMP=OR(ITEMP,NZ)-1
96          IF (IB.GT.36) IB=1
97          DAT(I)=FLOAT(ITEMP)
98      110      CONTINUE
99      IDL=250
100      C
101      RETURN
102      C
103      END

```

@PRT,S .CNVT1

```

LAMBERT*TPF$(O).SPHAVG(11)
1      SUBROUTINE SPHAVG (SPHX,SPHY)
2      DIMENSION X(1), X1(250), X9(250), Y1(250), Y9(250)
3      C
4      C THIS SUBROUTINE CALCULATES SPROCKET HOLE AVERAGES FOR EACH CAMERA
5      C FROM 5 CONTINUOUS FRAMES IN THE PHOTO DATA USING THE CHAIR OR SLED
6      C DISPLACEMENTS AND THE SPROCKET HOLE DISPLACEMENT.
7      C (THESE ARE THE 900 AND 100 RECORDS)
8      C
9      C ARGUMENT DEFINITIONS:
10     C   SPHX-RETURNED SPROCKET HOLE AVG. FOR X VALUES
11     C   SPHY-RETURNED SPROCKET HOLE AVG. FOR Y VALUES
12     C
13     GO TO 50
14     C
15     C
16     C SAVE X AND Y DISPLACEMENTS
17     C
18     ENTRY SPH9X (X)
19     DO 10 J=1,250
20     10  X9(J)=X(J)
21     ICX=ICX+1
22     RETURN
23     C
24     C
25     ENTRY SPH9Y (X)
26     DO 20 J=1,250
27     20  Y9(J)=X(J)
28     ICY=ICY+1
29     RETURN
30     C
31     C
32     ENTRY SPH1X (X)
33     DO 30 J=1,250
34     30  X1(J)=X(J)
35     ICX=ICX+1
36     RETURN
37     C
38     C
39     ENTRY SPH1Y (X)
40     DO 40 J=1,250
41     40  Y1(J)=X(J)
42     ICY=ICY+1
43     RETURN
44     C
45     C
46     50  CONTINUE
47     C
48     C CALCULATE DIFFERENCES IN SPROCKET HOLE AND SLED OR CHAIR
49     C TARGET DISPLACEMENTS FOR 5 CONSECUTIVE FRAMES,
50     C AND SUM THESE DIFFERENCES (X VALUES)
51     C
52     IBG=0
53     IF (ICX.LT.2) GO TO 80
54     C
55     SUMX=0
56     DO 70 J=1,250
57     IF (X1(J).GT.900..OR.X9(J).GT.900.) GO TO 60
58     SUMX=SUMX+(ABS(X1(J))-ABS(X9(J)))
59     IF (IBG.EQ.0) IBG=J

```

```

60          IF ((J-IBG).GE.4) GO TO 90
61          GO TO 70
62          SUMX=0
63          IBG=0
64          70  CONTINUE
65          C
66          IF (IBG.GE.250) IBG=0
67          C
68          80  SPHX=0
69          GO TO 100
70          90  CONTINUE
71          C
72          C
73          C  CALCULATE SPROCKET HOLE AVERAGES AND CONVERT TO PDS UNITS
74          C  FROM CENTIMETERS (X VALUE)
75          C
76          SPHX=(SUMX/5.)*(1000./(1.884765*.0254))
77          C
78          100 IF (ICV.LT.2) GO TO 120
79          C
80          C
81          C  CALCULATE DIFFERENCES IN SPROCKET HOLE AND SLED OR CHAIR
82          C  DISPLACEMENTS FOR 5 CONSECUTIVE FRAMES AND SUM THESE VALUES.
83          C  USE THE SAME 5 FRAMES USED IN CALCULATING THE X VALUES ABOVE
84          C  (Y VALUE)
85          C
86          IST=4
87          IF (IBG.NE.0) IST=IBG
88          IND=250
89          IF (IBG.NE.0) IND=IST+4
90          C
91          SUMY=0
92          DO 110 J=IST,IND
93          IF (Y1(J).GT.900..OR.Y9(J).GT.900.) GO TO 120
94          SUMY=SUMY+(ABS(Y1(J))-ABS(Y9(J)))
95          110 CONTINUE
96          C
97          GO TO 130
98          C
99          120 SPHY=0
100          GO TO 140
101          C
102          130 CONTINUE
103          C
104          C
105          C  CALCULATE SPROCKET HOLE AVERAGE AND CONVERT TO PDS UNITS
106          C  FROM CENTIMETERS (Y VALUE)
107          C
108          SPHY=(SUMY/5.)*(1000./(1.884765*.0254))
109          C
110          140 ICX=0
111          ICY=0
112          C
113          RETURN
114          C
115          END

```

OPRT,S .ASC8T9

LAMBERT*TPFS(O).ASC8T9(17)

```

1      SUBROUTINE ASC8T9(INPUTA,IWD,IBIT,NCHAR,CSTRNG)
2      C
3      C FUNCTION:
4      C   CONVERT STRING OF 8-BIT ASCII CHARACTERS TO
5      C   UNIVAC COMPATIBLE 9-BIT ASCII CHARACTER STRING
6      C
7      C ARGUMENT DEFINITIONS:
8      C   INPUTA-INTEGGER INPUT ARRAY OF 8-BIT CHARACTER DATA
9      C   IWD-STARTING WORD IN INPUTA OF 8-BIT ASCII CHARACTER STRING
10     C   IBIT-BEGINNING BIT IN IWD OF INPUT 8-BIT ASCII CHARACTER STRING
11     C   NCHAR-NUMBER ASCII CHARACTERS TO BE CONVERTED TO UNIVAC FORMAT
12     C   CSTRNG-RETURNED OUTPUT STRING OF UNIVAC 9-BIT ASCII CHARACTERS
13     C
14     C PROGRAMMER: J LAMBERT 20 MAY 1983
15     C
16     C INTEGER INPUTA(1)
17     C
18     C LENGTH OF CHARACTER STRING MUST BE INITIALIZED IN CALLING PROGRAM
19     C
20     C CHARACTER*(*) CSTRNG,ITEMP*1
21     C DATA IL1,IL2/2*0/
22     C IB=IBIT
23     C IW=IWD
24     C
25     C 1ST BIT OF UNIVAC ASCII CHARACTER ALWAYS 0(ZERO)
26     C
27     C BITS(ITEMP,1,1)=0
28     C
29     C CONVERT EACH 8-BIT CHARACTER TO 9-BITS
30     C
31     C DO 20,I=1,NCHAR
32     C IF((IB+7).LT.36)THEN
33     C   BITS(ITEMP,2,8)=BITS(INPUTA(IW),IB,8)
34     C ELSE
35     C   IL1=37-IB
36     C   IL2=8-IL1
37     C   BITS(ITEMP,2,IL1)=BITS(INPUTA(IW),IB,IL1)
38     C   BITS(ITEMP,2+IL1,IL2)=BITS(INPUTA(IW+1),1,IL2)
39     C END IF
40     C SUBSTR(CSTRNG,I,1)=SUBSTR(ITEMP,1,1)
41     C IB=IB+8
42     C IF(IB.GT.36)THEN
43     C   IW=IW+1
44     C   IB=IB-36
45     C END IF
46     C 20 CONTINUE
47     C RETURN
48     C END

```

@PRT,S .XQTPLT

LAMBERT*TPF\$(0).XQTPLT(11)

```

1  @XQT MPOOS1/PLOT
2  @.
3  @. CONTROL STATEMENT IMMEDIATELY AFTER @XQT STATEMENT
4  @. POS 5 - NO. INPUT UNITS (1, 2 OR 3)
5  @. POS 10 - TIME RECORD ANALYSIS PLOT OPTION
6  @. 0 = OUTPUT TO LINE PRINTER
7  @. 1 = OUTPUT TO COMPRESSED PLOT FILE
8  @. NOT 0 OR 1 = NO TIME RECORD ANALYSIS
9  @. POS 15 - XY CONTOUR PLOT OPTION
10 @. 0 = PLOT XY CONTOUR FOR ALL MOUNTS/SITES/RUNS
11 @. NOT 0 = NO XY CONTOUR PLOTS
12 @. POS 20 - FRAME PLOT OPTION
13 @. 0 = FRAME PLOTS FOR ALL MOUNTS/SITES/RUNS
14 @. NOT 0 = NO FRAME PLOTS
15 @. POS 25 - FORMAT OPTION FOR FRAME PLOTS
16 @. 0 = COL'S 2 & 4 UP -- COL'S 1, 3 & 5 DOWN
17 @. NOT 0 = ALL COL'S DOWN
18 @. POS 35 - LOWEST RUN NO. TO BE PROCESSED
19 @. POS 45 - HIGHEST RUN NO. TO BE PROCESSED
20 @XQT SCC$*INTLIB.FRBOPOP
21 MO40
22 DATASYSTEMS
23 5-4890
24 FRAME PLOTS
25 FRAME PLOTS
26 1 HARDCOPY 1 MICROFICHE
27
28
29 DRAW=1-END$
30 @DELETE,C COMPRESS.
31 @FREE COMPRESS.
32 @ASG,CP COMPRESS.,F///1000
33 @COPY 17.,COMPRESS.
34 @FREE COMPRESS.
35 @. EXITED FROM .XQTPLT

```

@PRT,S .PLOT

LAMBERT*TPF\$(0).PLOT(9)

```

1      C
2      C
3      C      FUNCTION:
4      C      THIS ASCII FORTRAN DRIVER PROGRAM CONTROLS THE GENERATION
5      C      OF MOUNT PLOTS FOR A RANGE OF RUNS SPECIFIED BY THE USER.
6      C      TWENTY-FIVE FRAMES (DATA PTS.) ARE PLOTTED ON EACH OF 10
7      C      PAGES FOR A GIVEN MOUNT AND SITE ID. THE DISSPLA PLOT
8      C      PACKAGE IS USED FOR ALL PLOTTING.
9
10     COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR.
11     *YOR,VARX,VARY,IERR,STEP,DC(250)
12     COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JOTDIG,DELTA,IPGVER
13     CHARACTER*6 RUNID,VARID,JOTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
14     CHARACTER*6 FRUN,LRUN,RUNS(200),FIRST,LAST,ENC*4,SITE*1
15     DATA MAX/280/
16
17     C
18     READ USER OPTIONS IN CARD IMAGE
19
20     C
21     READ(5,6) IFILE,ITIM,IXY,IMTPLT,IWAY,FRUN,LRUN
22     FORMAT(5I5,9X,2(A6,4X))
23
24     C
25     WHERE:
26     IFILE: NO. OF INPUT FILES
27     ITIM: TIME RECORD ANALYSIS PLOT OPTION
28     0=OUTPUT TO LINE PRINTER
29     1=OUTPUT TO COMPRESSED PLOT FILE
30     NOT 0 AND NOT 1 = NO TIME RECORD ANALYSIS
31     IXY: XY CONTOUR PLOT OPTION
32     0=PLOT XY CONTOUR FOR EACH MOUNT/SITE/RUN
33     NOT 0 = NO XY CONTOURS
34     IMTPLT: MOUNT PLOT OPTION
35     0=FRAME PLOTS REQUIRED FOR ALL DATA
36     NOT 0 = NO FRAME PLOTS
37     IWAY: PLOT DIRECTION OPTION
38     0=BOTTOM TO TOP COLUMNS 2 AND 4
39     NOT 0 = TOP TO BOTTOM ALL COLUMNS
40     FRUN: LOW ORDER VALUE OF RUN NO. RANGE
41     LRUN: HIGH ORDER VALUE OF RUN NO. RANGE
42
43     C
44     SET COMPRESSOR FOR PLOTTING
45
46     C
47     CALL COMPRS
48     CALL SETDEV(6,0)
49     CALL GRACE(0.)
50     CALL PAGE(14./1.02, 1./1.02)
51     CALL NOBRDR
52     CALL ERASE
53
54     C
55     ONE UNIT = 10.5/1.02 INCHES
56
57     C
58     CALL UNITS(10.5/1.02)
59
60     C
61     COMPUTE RUN RANGE
62
63     C
64     FIRST=SUBSTR(FRUN,3,4)
65     LAST=SUBSTR(LRUN,3,4)
66     DECODE(4,FIRST)IFIRST
67     DECODE(4,LAST)ILAST
68     FORMAT(I4)

```

```

60      C
61      C      COMPUTE INPUT UNIT ASSIGNMENTS
62      C
63      ILIM=IFILE+8
64      IU=9
65      1      CONTINUE
66      PRINT 3,IU,ILIM,ITIM,IXY,IMTPLT,IWAY,FRUN,LRUN
67      3      FORMAT(6I10,2(4X,A6))
68      C
69      C      GENERATE TITLE FOR EACH PAGE
70      C
71      TITLE(2)='    CAM'
72      TITLE(5)='    '
73      TITLE(6)='    PAG'
74      TITLE(7)='E    '
75      TITLE(9)='    '
76      TITLE(3)='    '
77      TITLE(4)='    '
78      TITLE(8)='    '
79      C
80      C      PLOT DATA FOR EACH RUN
81      C
82      IRN=IABS(ILAST-IFIRST)+1
83      INC=IFIRST
84      DO 500,I=1,IRN
85      C
86      C      CREATE RUN ARRAY TO BE SEARCHED
87      C
88      SUBSTR(RUNS(I),1,2)='LX'
89      ENCODE(17,ENC)INC
90      17      FORMAT(I4)
91      SUBSTR(RUNS(I),3,4)=ENC
92      INC=INC+1
93      C
94      C      ATTEMPT TO SET SECTOR TO HEADER FOR RUN
95      C
96      CALL SETSCT(IU,RUNS(I),'PHDATA',1,ISITE,NREC,IER)
97      IF(IER.NE.0)GO TO 500
98      CALL GETRCD(IU,MAX,RUNID,VARID,ISITE,IDL,DC,IER)
99      IF(IER.NE.0)THEN
100         PRINT 13,IER
101         13      FORMAT(1X,'STATUS=',03)
102         STOP 'HEADER ERROR'
103      END IF
104      IF(VARID.NE.'PHDATA')STOP 'HEADER ERROR'
105      C
106      C      GENERATE TIME RECORD ANALYSIS AND WRITE ON FICHE
107      C
108      ENCODE(10,SITE)ISITID
109      10      FORMAT(I1)
110      SUBSTR(TITLE(3),3,1)=SITE
111      TITLE(1)=RUNID
112      IF(ITIM.EQ.0 .OR. ITIM.EQ.1)THEN
113         CALL TIMER(IU,ITIM)
114         IF(IERR.NE.0)GO TO 500
115      END IF
116      C
117      C      PLOT DATA FOR EACH MOUNT FOR THIS RUN
118      C
119      DO 400,L=1,4

```

```

120      C
121      C      LOAD DATA
122      C
123      CALL LDATA(IU,L)
124      IF(IX.NE.O)THEN
125          IST=1
126          IFM=1
127      ELSE
128          GO TO 400
129      END IF
130      C
131      C      GENERATE TITLE INFORMATION FOR THIS MOUNT
132      C
133      IF(L.EQ.1)THEN
134          TITLE(5)='NECK '
135      ELSE IF(L.EQ.2)THEN
136          TITLE(5)='MOUTH '
137      ELSE IF(L.EQ.3)THEN
138          TITLE(5)='HEAD '
139      ELSE IF(L.EQ.4)THEN
140          TITLE(5)='PELVIS'
141      END IF
142      C
143      C      LOOP FOR EACH PAGE OF PLOTS (FRAMES)
144      IF(IMTPLT.NE.O)GO TO 300
145      C
146      DO 55,IPGE=1,10
147      C
148      C      DEFINE PHYSICAL ORIGIN FOR FIRST FRAME ON PAGE
149      C
150      XOR=.16
151      YOR=.81
152      C
153      C      DETERMINE MAX AND MIN VALUES FOR THIS PAGE OF PLOTS
154      C
155      ILT=IST+24
156      CALL MIXY
157      C
158      C      PRINT TITLE AT TOP OF PAGE
159      C
160      CALL PLOT1
161      C
162      C      LOOP TO CONTROL COLUMN OF PLOTS
163      C
164      DO 50,K=1,5
165      IC=MOD(K,2)
166      C
167      C      PLOT DATA 5 FRAMES FOR EACH OF 5 COLUMNS
168      C
169      DO 25,J=1,5
170      CALL PLOT2(IFM)
171      C
172      C      CHECK FOR DIRECTION OF PLOTS (UP OR DOWN)
173      C
174      IF(IWAY.NE.O)THEN
175      C
176      C      PLOT TOP TO BOTTOM ALL COLUMNS
177      C
178      IF(J.NE.5)YOR=YOR-.2
179      ELSE

```



```

180      C
181      C      PLOT BOTTOM TO TOP COLUMNS 2 AND 4
182      C
183          IF(J.NE.5)THEN
184              IF(IC.NE.0)THEN
185                  YOR=YOR-.2
186              ELSE
187                  YOR=YOR+.2
188              END IF
189          END IF
190      END IF
191      C
192      C      CHECK FOR LAST DATA POINT
193      C
194          IF(IST.LT.250)THEN
195              IST=IST+1
196              IFM=IFM+1
197          ELSE
198              CALL ENOPL(O)
199              GO TO 55
200          END IF
201      25  CONTINUE
202      C
203      C      MODIFY PHYSICAL ORIGIN FOR NEXT COLUMN
204      C
205          IF(IWAY.NE.0)THEN
206              XOR=XOR+.2
207              YOR=.81
208          ELSE
209              XOR=XOR+.2
210          END IF
211      50  CONTINUE
212          CALL ENOPL(O)
213      55  CONTINUE
214      C
215      C      PLOT X VS Y ALL FRAMES
216      C
217      300 IF(IXY.EQ.0)CALL XYPLOT
218      400 CONTINUE
219      500 CONTINUE
220          IF(IU.LT.ILIM)THEN
221              IU=IU+1
222              GO TO 1
223          END IF
224          CALL DONEPL
225          STOP
226          END

```

◆PRT,S .PLOTD

```

LAMBERT=TPFS(0).TIMER(33)
1      SUBROUTINE TIMER(IU,ITIM)
2
3      C
4      C      FUNCTION:
5      C      GENERATE AND PLOT TIME RECORD ANALYSIS FOR A GIVEN RUN
6
7      C      ARGUMENT DEFINITIONS:
8      C      IU: LOGICAL UNIT NO. OF INPUT FILE
9      C      ITIM: OPTION TO PLOT TIME RECORD ANALYSIS WHERE
10     C      0 = OUTPUT TO LINE PRINTER
11     C      1 = OUTPUT TO COMPRESSED PLOT FILE
12
13     C      COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
14     C      *YOR,VARX,VARY,IERR,STEP,DC(250)
15     C      COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
16     C      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
17     C      CHARACTER*6 SYM*1(250),LTINT,LDD*9(250),LETIME*9(250)
18     C      CHARACTER*8 LINDEX*3(250),LPDST,LEXPT,LFRNO*7
19     C      CHARACTER*128 LINE,LHEAD(2)
20     C      DIMENSION ETIME(250),DD(250)
21     C      DATA LPDST,LEXPT,LFRNO/'D TIME','P TIME','FRAME '/
22     C      IERR=1
23     C      CALL HEIGHT(.018)
24
25     C
26     C      SET SECTOR FOR TIME RECORD
27
28     C      VARID=' TS '
29     C      SUBSTR(VARID,1,1)=SUBSTR(TITLE(3),3,1)
30     C      CALL SETSCT(IU,RUNID,VARID,1,ISIT,NREC,IERR)
31     C      IF(IERR.NE.0)THEN
32     C          PRINT 13,IERR
33     C          FORMAT(1X,'STATUS=',03)
34     C          GO TO 100
35     C      END IF
36
37     C
38     C      READ TIME RECORD
39
40     C      MAX=280
41     C      CALL GETRCD(IU,MAX,RUNID,VARID,ISIT,IDL,DD,IERR)
42     C      IF(IERR.NE.0)THEN
43     C          IF(IERR.EQ.1)THEN
44     C              GO TO 100
45     C          ELSE
46     C              PRINT 13,IERR
47     C              STOP 'TIME ERROR'
48     C          END IF
49     C      END IF
50
51     C
52     C      CHECK FOR TIME RECORD IDENTIFICATION
53
54     C      IF(SUBSTR(VARID,2,2).NE.'TS')THEN
55     C          PRINT 1,RUNID
56     C          FORMAT(1X,'COULD NOT FIND TIME RECORD FOR RUN ',A6)
57     C          STOP 'TIME ERROR'
58     C      ELSE
59     C          IERR=0
60     C      END IF
61
62     C
63     C      COMPUTE TIME INTERVAL BETWEEN FRAMES

```

```

60      DO 15,K=250,1,-1
61      IF(DD(K).LE.O)THEN
62          CONTINUE
63      ELSE
64          DO 21,J=1,250
65          IF(DD(J).LE.O)THEN
66              CONTINUE
67          ELSE
68              IF(DD(K).LT.DD(J))DD(K)=DD(K)+10.
69              TINT=(DD(K)-DD(J))/(K-J)
70              GO TO 32
71          END IF
72      21  CONTINUE
73          GO TO 100
74      END IF
75      15  CONTINUE
76      PRINT 16,RUNID
77      16  FORMAT(1X,'TIME RECORD FOR RUN ',A6,' WAS ALL ZEROES')
78      GO TO 100
79      32  ETIME(1)=DD(1)
80          SYM(1)=' '
81      C
82      C      COMPUTE EXPECTED TIMES
83      C
84      DO 35,JJ=2,250
85      ETIME(JJ)=ETIME(JJ-1)+TINT
86      SYM(JJ)=' '
87      IF(DD(JJ).LT..000001)THEN
88          CONTINUE
89      ELSE
90          DIF=ETIME(JJ)-DD(JJ)
91          IF(ABS(DIF).GT..001)SYM(JJ)='*'
92      END IF
93      35  CONTINUE
94      C
95      C      WRITE HEADER INFO FOR TIME RECORD ANALYSIS
96      C
97      ENCODE(11,LTINT)TINT
98      11  FORMAT(F6.5)
99      IF(ITIM.EQ.1)THEN
100          CALL TABLET('CENTER','LONG')
101      ELSE IF(ITIM.EQ.O)THEN
102          PRINT 56
103      56  FORMAT('1')
104      END IF
105      SUBSTR(LHEAD(1),1,4)='RUN '
106      SUBSTR(LHEAD(1),5,6)=RUNID
107      SUBSTR(LHEAD(1),11,11)=' '
108      SUBSTR(LHEAD(1),22,7)='CAMERA '
109      SUBSTR(LHEAD(1),29,3)=SUBSTR(TITLE(3),1,3)
110      SUBSTR(LHEAD(1),32,11)=' '
111      SUBSTR(LHEAD(1),43,15)='PREDICTED RATE '
112      SUBSTR(LHEAD(1),58,6)=LTINT
113      IF(ITIM.EQ.1)THEN
114          SUBSTR(LHEAD(1),64,1)='$'
115          CALL CTLINE(LHEAD(1))
116          CALL CTLINE(' $')
117      ELSE IF(ITIM.EQ.O)THEN
118          PRINT 14,LHEAD(1)
119      14  FORMAT(21X,A63/)

```

```

120      END IF
121      C
122      C   WRITE HEADER INFO. FOR INDIVIDUAL COLUMNS
123      C
124      CALL HEIGHT(.01)
125      DO 23,IREP=1,5
126      SUBSTR(LHEAD(2),(((IREP-1)*23)+1),7)=LFRNO
127      SUBSTR(LHEAD(2),(((IREP-1)*23)+8),8)=LPDST
128      SUBSTR(LHEAD(2),(((IREP-1)*23)+16),8)=LEXPT
129      23  CONTINUE
130      IF(ITIM.EQ.1)THEN
131      SUBSTR(LHEAD(2),116,1)='$'
132      CALL CTLINE(LHEAD(2))
133      CALL CTLINE('$')
134      ELSE IF(ITIM.EQ.0)THEN
135      PRINT 17,LHEAD(2)
136      17  FORMAT(1X,A115/)
137      END IF
138      C
139      C   PRINT TIME RECORD ANALYSIS
140      C
141      DO 60,KK=1,50
142      DO 71,IENC=1,5
143      INDEX=(((IENC-1)*50)+KK)
144      ENCODE(8,LINDEX(INDEX))INDEX
145      ENCODE(9,LDD(INDEX)) DD(INDEX)
146      ENCODE(9,LETIME(INDEX)) ETIME(INDEX)
147      9  FORMAT(F9.5)
148      8  FORMAT(I3)
149      SUBSTR(LINE,(((IENC-1)*23)+1),1)=' '
150      SUBSTR(LINE,(((IENC-1)*23)+2),3)=LINDEX(INDEX)
151      SUBSTR(LINE,(((IENC-1)*23)+5),9)=LDD(INDEX)
152      SUBSTR(LINE,(((IENC-1)*23)+14),9)=LETIME(INDEX)
153      SUBSTR(LINE,(((IENC-1)*23)+23),1)=SYM(INDEX)
154      71  CONTINUE
155      IF(ITIM.EQ.1)THEN
156      SUBSTR(LINE,116,1)='$'
157      CALL CTLINE(LINE)
158      ELSE IF(ITIM.EQ.0)THEN
159      PRINT 18,LINE
160      18  FORMAT(1X,A115)
161      END IF
162      60  CONTINUE
163      IF(ITIM.EQ.0)THEN
164      PRINT 78
165      78  FORMAT('1')
166      END IF
167      100 CONTINUE
168      CALL RESET('HEIGHT')
169      IF(ITIM.EQ.1)THEN
170      CALL ENDTAB(0)
171      CALL ENDPL(0)
172      END IF
173      RETURN
174      END

```

•PRT,S .XOTS GN

```

LAMBERT*TPF$(O).LDATA(13)
1      SUBROUTINE LDATA(IU,MNT)
2
3      C
4      C      FUNCTION:
5      C      LOAD PHOTO DISPLACEMENT DATA INTO COMMON FOR
6      C      ACCESS BY PLOT ROUTINES
7
8      C      ARGUMENT DEFINITIONS:
9      C      IU: LOGICAL UNIT NO. OF INPUT DATA FILE
10     C      MNT: MOUNT INDEX
11
12     C
13     C      COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
14     C      *YOR,VARX,VARY,IERR,STEP,DC(250),DX(250,12),DY(250,12)
15     C      COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
16     C      DIMENSION DD(250)
17     C      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
18     C      CHARACTER*2 ONE/'01'/,TGCHK/'00'/,MTCHK*1/'0'/
19     C      CHARACTER*3 TARGET,OLDRUN*6
20     C      DATA MAX/280/
21     C      IX=0
22     C      IY=0
23
24     C
25     C      INITIALIZE CHAIR DATA FLAG
26     C
27     C      IF(OLDRUN.NE.RUNID)ICHAIR=0
28
29     C
30     C      SET SECTOR TO TIME RECORD
31     C
32     C      VARID=' TS '
33     C      SUBSTR(VARID,1,1)=SUBSTR(TITLE(3),3,1)
34     C      CALL SETSCT(IU,RUNID,VARID,1,ISIT,NREC,IER)
35     C      IF(IER.NE.0)THEN
36     C          PRINT 13,IER
37     C          FORMAT(1X,'STATUS='',03)
38     C          STOP 'LDATA'
39     C      END IF
40
41     C
42     C      GET NEXT DATA RECORD
43     C
44     C      CALL GETRCD(IU,MAX,RUNID,VARID,ISIT,IDL,DD,IER)
45     C      IF(IER.NE.0)THEN
46     C          IF(IER.EQ.1)THEN
47     C              GO TO 56
48     C          ELSE
49     C              PRINT 13,IER
50     C              GO TO 72
51     C          END IF
52     C      END IF
53
54     C
55     C      CHECK FOR CHAIR DATA AND STORE
56     C
57     C      IF(ICHAIR.EQ.1)THEN
58     C          CONTINUE
59     C      ELSE
60     C          TARGET=SUBSTR(VARID,3,3)
61     C          IF(TARGET.EQ.'100')THEN
62     C              DO 16,J=1,250
63     C                  DC(J)=DD(J)
64     C                  PRINT 606,VARID,RUNID
65     C                  FORMAT(' CHAIR DATA, VARIABLE',A6,' LOADED FOR RUN ',A6)
66     C          END IF
67     C      END IF

```

```

60          ICHAIR=1
61          GO TO 8
62      END IF
63  END IF
64      C
65      C      PROCESS DATA RECORD
66      C
67      TGCHK=SUBSTR(VARID,4,2)
68      IF(TGCHK.LT.ONE)THEN
69          CONTINUE
70      ELSE
71          MTCHK=SUBSTR(VARID,3,1)
72          DECODE(10,MTCHK)IMTCHK
73      10      FORMAT(I1)
74      C
75      C      CHECK FOR CURRENT MOUNT
76      C
77          IF(IMTCHK.NE.MNT)THEN
78              CONTINUE
79          ELSE
80              IF(SUBSTR(VARID,2,1).EQ.'X')THEN
81                  IX=IX+1
82                  DO 22,J=1,250
83      22          DX(J,IX)=DD(J)
84                  VARX(IX)=VARID
85              ELSE IF(SUBSTR(VARID,2,1).EQ.'Y')THEN
86                  IY=IY+1
87                  DO 32,J=1,250
88      32          DY(J,IY)=DD(J)
89                  VARY(IY)=VARID
90              END IF
91          END IF
92      END IF
93      C
94      C      READ ANOTHER DATA RECORD
95      C
96      GO TO 8
97      56      CONTINUE
98          IF(IY.EQ.0)THEN
99              IX=0
100          ELSE IF(IX.EQ.0)THEN
101              CONTINUE
102          ELSE
103              PRINT 58,IX,IY,MNT,RUNID
104      58          FORMAT(1X,I2,2X,'X DISP AND ',I2,2X,'Y DISP',
105          *' LOADED FOR MOUNT ',I2,5X,'RUN ',A6)
106              PRINT 66,(VARX(I),I=1,IX),(VARY(I),I=1,IY)
107      66          FORMAT(12(2X,A6))
108          END IF
109      C
110      C      IF NO CHAIR DATA STORE 999'S IN ARRAY
111      C
112          IF(ICAIR.EQ.0)THEN
113              DO 71,J=1,250
114      71          DC(J)=999.
115          END IF
116      72      CONTINUE
117      OLDRUN=RUNID
118      RETURN
119      END

```

```

LAMBERT=TPFS(O).MIXY(30)
1      SUBROUTINE MIXY
2      C
3      C      FUNCTION:
4      C      DETERMINE MINIMUM AND MAXIMUM X AND Y VALUES FOR
5      C      A PAGE OF MOUNT PLOTS
6      C
7      COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
8      *YOR,VARX,VARY,IERR,STEP,DC(250),DX(250,12),DY(250,12)
9      COMMON/PTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
10     CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
11     XMIN,YMIN=999.
12     XMAX,YMAX=0.
13     C
14     C      INITIALIZE COUNTERS
15     C
16     ISTNEW=IST
17     ILTNEW=ILT
18     C
19     C      CHECK IF ALL PTS TO BE PLOTTED
20     C
21     IF(IERR.EQ.999)THEN
22         ISTNEW=1
23         ILTNEW=250
24     END IF
25     C
26     C      EXAMINE DATA PTS. FOR ALL TARGETS
27     C
28     DO 20,JK=ISTNEW,ILTNEW
29     DO 20,JL=1,IX
30     DXJ=ABS(DX(JK,JL))
31     DYJ=ABS(DY(JK,JL))
32     IF(DXJ.LE.990. .AND. DYJ.LE.990.)THEN
33         XMIN=AMIN1(DXJ,XMIN)
34         YMIN=AMIN1(DYJ,YMIN)
35         XMAX=AMAX1(DXJ,XMAX)
36         YMAX=AMAX1(DYJ,YMAX)
37     END IF
38     20 CONTINUE
39     C
40     C      EQUATE X AND Y AXES
41     C
42     XDIF=XMAX-XMIN
43     YDIF=YMAX-YMIN
44     IF(XDIF.LE.YDIF)THEN
45         XMAX=XMAX+(YDIF-XDIF)/2.
46         XMIN=XMIN-(YDIF-XDIF)/2.
47     ELSE
48         YMAX=YMAX+(XDIF-YDIF)/2.
49         YMIN=YMIN-(XDIF-YDIF)/2.
50     END IF
51     C
52     C      COMPUTE STEP INTERVAL FOR PLOT
53     C
54     IF(IERR.NE.999)THEN
55         STEP=(XMAX-XMIN)/.18
56     ELSE
57         STEP=(XMAX-XMIN)/.8
58     END IF
59     RETURN

```

60

END

●PRT, S .LDATA


```

LAMBERT*TPF$(0).PLOTD(46)
1      SUBROUTINE PLOTD
2      C
3      C      FUNCTION:
4      C      CONTROLS (1) GENERATION OF TITLE FOR EACH PAGE OF MOUNT
5      C      PLOTS AND (2) GENERATION OF MOUNT PLOTS WITH DISSPLA
6      C      PLOT PACKAGE
7      C
8      COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
9      *YOR,VARX,VARY,IERR,STEP,DC(250),DX(250,12),DY(250,12)
10     COMMON/PHTHDR/RUNID,VARID,ISITID,FITIME,JOTDIG,DELTA,IPGVER
11     CHARACTER*6 RUNID,VARID,JOTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
12     *,PAG*3
13     CHARACTER*2 TGT(12),TADD,TDROP,TNEW(12),TOLD(12),SSCL*8,SC*7
14     CHARACTER*4 FRAM/'  '/,TEMP*2/'  '/
15     DIMENSION DAT(12,2),NPT(12)
16     C
17     C      PLOT TITLE AT TOP OF FRAME
18     C
19     C
20     ENTRY PLOT1
21     C
22     C      FUNCTION:
23     C      GENERATE TITLE INFO. FOR PAGE OF MOUNT PLOTS
24     C
25     CALL RESET('PHYSOR')
26     CALL AREA2D(1,..8)
27     CALL HEIGHT(.018)
28     C
29     C      PRINT X-SCALE VALUE FOR THIS PAGE OF PLOTS (METERS)
30     C
31     SCAL=XMAX-XMIN
32     ENCODE(11,SC)SCAL
33     11  FORMAT(F7.6)
34     SUBSTR(SSCL,1,7)=SC
35     SUBSTR(SSCL,7,1)='$'
36     CALL MESSAG(SSCL,100,..39,..925)
37     C
38     C      PLOT TITLE
39     C
40     ENCODE(3,PAG)IPGE
41     SUBSTR(TITLE(7),3,3)=PAG
42     3   FORMAT(I3)
43     CALL HEIGHT(.027)
44     CALL MESSAG(TITLE,44,0,..92)
45     CALL ENDGR(0)
46     CALL RESET('HEIGHT')
47     RETURN
48     C
49     C
50     ENTRY PLOT2(IFM)
51     C
52     C      FUNCTION:
53     C      GENERATE MOUNT PLOT FOR A FRAME OF DATA
54     C      ARGUMENT DEFINITION:
55     C      IFM: CURRENT FRAME COUNT
56     C
57     C      PLOT DATA POINTS
58     C
59     CALL HEIGHT(.01)

```

```

60      CALL PHYSOR(XOR,YOR)
61      CALL AREA2D(.18,.18)
62      C
63      C   WRITE BLOCK NO. CORRESPONDING TO DATA PT.
64      C
65      XFM=.153
66      YFM=.165
67      ENCODE(7,FRAM)IFM
68      7   FORMAT(I3)
69      SUBSTR(FRAM,4,1)='$'
70      CALL MESSAG(FRAM,100,XFM,YFM)
71      C
72      C   PLOT SYMBOL FOR CHAIR DATA (IF CHAIR DATA AVAILABLE)
73      C
74      IF(DC(IST).GT.990)THEN
75          CONTINUE
76      ELSE
77          XCH=.005
78          YCH=.165
79          CALL MESSAG('C',1,XCH,YCH)
80      END IF
81      C
82      C   SCALE DATA TO PRESENT PLOTTING AREA
83      C
84      CALL GRAF(XMIN,STEP,XMAX,YMIN,STEP,YMAX)
85      C
86      C   DRAW BLOCK AROUND SUBPLOT
87      C
88      CALL RLVEC(XMIN,YMIN,XMIN,YMAX,0000)
89      CALL RLVEC(XMAX,YMAX,XMIN,YMAX,0000)
90      CALL RLVEC(XMAX,YMAX,XMAX,YMIN,0000)
91      CALL RLVEC(XMIN,YMIN,XMAX,YMIN,0000)
92      NTHIS=0
93      DO 50,J=1,IX
94          NPT(J)=0
95          IF(DX(IST,J).GT.990.OR.DY(IST,J).GT.990)THEN
96              CONTINUE
97          ELSE
98              NTHIS=NTHIS+1
99      C
100     C   LOAD DATA FOR ALL TARGETS IN THIS FRAME
101     C
102         DAT(NTHIS,1)=ABS(DX(IST,J))
103         DAT(NTHIS,2)=ABS(DY(IST,J))
104     C
105     C   IDENTIFY MANUAL ENTRIES
106     C
107         IF(DX(IST,J).LT.0.)NPT(NTHIS)=1
108         TGT(NTHIS)=SUBSTR(VARX(J),4,2)
109         IF(SUBSTR(VARX(J),4,2).NE.SUBSTR(VARY(J),4,2))THEN
110             PRINT 52
111             52   FORMAT(' X + Y DISP. TARGETS DO NOT AGREE ')
112             STOP 'PLOTD'
113         END IF
114     END IF
115     50   CONTINUE
116     IF(NTHIS.NE.0)THEN
117         DO 60,J=1,NTHIS
118             X=XPSN(DAT(J,1),DAT(J,2))
119             Y=YPSN(DAT(J,1),DAT(J,2))

```

```

120         IF(TGT(J).LE.'09')THEN
121             XT=X
122             YT=Y
123             XM=X-.004
124             YM=Y-.009
125             SUBSTR(TEMP,1,1)=SUBSTR(TGT(J),2,1)
126             IT=1
127         ELSE
128             XT=X-.006
129             YT=Y
130             XM=X-.007
131             YM=Y-.009
132             TEMP=TGT(J)
133             IT=2
134         END IF
135     C
136     C      PLOT TARGET
137     C
138         CALL MESSAG(TEMP,IT,XT,YT)
139     C
140     C      PLOT CIRCLE AROUND TARGET IF MANUAL ENTRY
141     C
142         IF(NPT(J).NE.0)THEN
143             CALL HEIGHT(.029)
144             CALL INTNO(0,XM,YM)
145             CALL HEIGHT(.01)
146         END IF
147     60     CONTINUE
148         INS=NTHIS-1
149     C
150     C      CONNECT TARGETS WITH LINES
151     C
152         DO 68,J=1,INS
153             J1=J+1
154             DO 68,J2=J1,NTHIS
155                 CALL RLVEC(DAT(J,1),DAT(J,2),DAT(J2,1),DAT(J2,2),0000)
156     68     CONTINUE
157     END IF
158     TADD=' '
159     TDROP=' '
160     C
161     C      DETERMINE WHICH TARGETS HAVE BEEN ADDED OR DELETED FROM LAST FRAME
162     C
163         IF(IST.NE.1)THEN
164             IF(NTHIS.NE.0.OR.NLAST.NE.0)THEN
165                 IF(NTHIS.NE.0)THEN
166                     DO 112,J=1,NTHIS
167                         TNEW(J)=TGT(J)
168                     IF(NLAST.NE.0)THEN
169                         DO 113,J=1,NLAST
170                             DO 113,K=1,NTHIS
171                                 IBOTH=0
172                                 IF(TOLD(J).EQ.TNEW(K))IBOTH=1
173                                 IF(IBOTH.EQ.1)THEN
174                                     TOLD(J)=' '
175                                     TNEW(K)=' '
176                                 END IF
177                     CONTINUE
178                     DO 114,J=1,NLAST
179                         IF(TOLD(J).NE.' ')TDROP=TOLD(J)

```

```

180         END IF
181         DO 115,K=1,NTHIS
182           115 IF(TNEW(K).NE.' ')TADD=TNEW(K)
183           ELSE IF(NTHIS.NE.O)THEN
184             IF(NLAST.EQ.O)THEN
185               DO 214,J=1,NLAST
186                 214 IF(TOLD(J).NE.' ')TDROP=TOLD(J)
187             END IF
188           END IF
189           ELSE IF(NTHIS.EQ.O .AND. NLAST.EQ.O)THEN
190             GO TO 111
191           END IF
192           ELSE IF(IST .EQ. 1)THEN
193             IF(NTHIS.EQ.O)GO TO 111
194           END IF
195         C
196         C   STORE TARGET NOS. FOR THIS FRAME
197         C
198         DO 116,J=1,NTHIS
199           116 TOLD(J)=TGT(J)
200           NLAST=NTHIS
201         C
202         C   PLOT SYMBOL IF NEW TARGET WAS VISIBLE IN THIS FRAME
203         C
204         XNEW=.005
205         YNEW=.007
206         IF(TADD.LE.'09')THEN
207           SUBSTR(TEMP,1,1)=SUBSTR(TADD,2,1)
208           IA=1
209         ELSE
210           TEMP=TADD
211           IA=2
212         END IF
213         CALL MESSAG(TEMP,IA,XNEW,YNEW)
214         C
215         C   PLOT SYMBOL IF A TARGET WAS DROPPED IN THIS FRAME
216         C
217         XDROP=.159
218         YDROP=.007
219         IF(SUBSTR(TDROP,1,1).EQ.'O')SUBSTR(TDROP,1,1)=' '
220         CALL MESSAG(TDROP,2,XDROP,YDROP)
221       111 CONTINUE
222         C
223         C   END SUBPLOT
224         C
225         CALL RESET('HEIGHT')
226         CALL ENDGR(O)
227         RETURN
228         END

```

•PRT,S .MIXY

```

LAMBERT=IPFS(0).XYPLOT(22)
1      SUBROUTINE XYPLOT
2      C
3      C      FUNCTION:
4      C      PLOT X AND Y DISPLACEMENT VALUES FOR GIVEN RUN AND ANATOMICAL
5      C      MOUNT; EACH PHOTO TARGET PLOTTED ON SAME GRAPH
6      C
7      DIMENSION DXARY(250),DYARY(250)
8      COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
9      *YOR,VARX,VARY,IERR,STEP,DC(250),DX(250,12),DY(250,12)
10     COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
11     CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
12     CHARACTER*6 XYTIT(9),TAR*2,TIME*8,DATE*8
13     DATA NPTS,ITAR/2*0/
14     C
15     C      LOAD TITLE INFO.
16     C
17     DO 12,I=1,9
18     XYTIT(I)=TITLE(I)
19     12 CONTINUE
20     XYTIT(5)=' '
21     XYTIT(6)=' '
22     XYTIT(7)=' '
23     XYTIT(8)=' '
24     XYTIT(9)=' '
25     CALL ADATE(DATE,TIME)
26     XYTIT(4)=TITLE(5)
27     SUBSTR(XYTIT(5),4,2)=SUBSTR(DATE,1,2)
28     SUBSTR(XYTIT(6),1,2)=SUBSTR(DATE,3,2)
29     SUBSTR(XYTIT(6),4,2)=SUBSTR(DATE,5,2)
30     C
31     C      SUPPRESS LISTING FOR OUT OF RANGE POINTS
32     C
33     CALL NOCHEK
34     CALL RESET('PHYSOR')
35     C
36     C      LABEL AXES
37     C
38     CALL AREA2D(.8,.8)
39     CALL XNAME(' FILM PLANE X DISPLACEMENT (METERS)',35)
40     CALL YNAME(' FILM PLANE Y DISPLACEMENT (METERS)',35)
41     C
42     C      PRINT HEADING
43     C
44     CALL HEIGHT(.027)
45     CALL MESSAG(XYTIT,50,0...92)
46     CALL HEIGHT(.016)
47     C
48     C      COMPUTE MIN AND MAX VALUES FOR X AND Y
49     C
50     IERR=999
51     CALL MIXY
52     IERR=0
53     C
54     C      SCALE DATA FOR PLOTS
55     C
56     CALL GRAF(XMIN,STEP,XMAX,YMIN,STEP,YMAX)
57     C
58     C      PLOT X-Y CONTOUR FOR EACH TARGET
59     C

```

```

60      CALL HEIGHT(.022)
61      DO 10,I=1,IX
62      LINE=MOD(I,5)
63      IF(LINE.EQ.0)THEN
64          CALL DOT
65      ELSE IF(LINE.EQ.1)THEN
66          CALL RESET('DOT')
67      ELSE IF(LINE.EQ.2)THEN
68          CALL CHNDOT
69      ELSE IF(LINE.EQ.3)THEN
70          CALL DASH
71      ELSE IF(LINE.EQ.4)THEN
72          CALL CHNDOSH
73      END IF
74      C
75      C      LOAD DATA INTO PLOT ARRAY AND CHANGE NEGATIVE SIGNS
76      C
77      DO 30,J=1,250
78      DXARY(J)=DX(J,I)
79      DYARY(J)=DY(J,I)
80      IF(DXARY(J).LT.0.)THEN
81          DXARY(J)=-1.*DXARY(J)
82      END IF
83      IF(DYARY(J).LT.0.)THEN
84          DYARY(J)=-1.*DYARY(J)
85      END IF
86      30  CONTINUE
87      C
88      C      PLOT DATA (ONLY NON-ZERO PTS)
89      C
90      IPOS=1
91      ITAR=0
92      5   DO 35,J=IPOS,250
93      IF(DXARY(J).GT.990..OR.DYARY(J).GT.990.)THEN
94          CONTINUE
95      ELSE
96          NPTS=0
97          IPLOT=J
98          GO TO 15
99      END IF
100     35  CONTINUE
101     GO TO 10
102     15  DO 40,K=IPLOT,250
103     IF(DXARY(K).GT.990..OR.DYARY(K).GT.990.)THEN
104         IF(ITAR.EQ.0)THEN
105             SX=XPOSN(DXARY(IPLOT),DYARY(IPLOT))
106             SY=YPOSN(DXARY(IPLOT),DYARY(IPLOT))
107             TAR=SUBSTR(VARX(1),4,2)
108             IF(SUBSTR(TAR,1,1).EQ.'O')SUBSTR(TAR,1,1)=' '
109             CALL MESSAG(TAR,2,SX,SY)
110             ITAR=1
111         END IF
112         CALL CURVE(DXARY(IPLOT),DYARY(IPLOT),NPTS,0)
113         IPOS=K+1
114         IF(K.LT.250)GO TO 5
115     ELSE
116         NPTS=NPTS+1
117         IF(K.EQ.250)THEN
118             IF(ITAR.EQ.0)THEN
119                 SX=XPOSN(DXARY(IPLOT),DYARY(IPLOT))

```

```

120          SY=YPOSN(DXARY(IPLT),DYARY(IPLT))
121          TAR=SUBSTR(VARX(1),4,2)
122          IF(SUBSTR(TAR,1,1).EQ.'O')SUBSTR(TAR,1,1)=' '
123          CALL MESSAG(TAR,2,SX,SY)
124          ITAR=1
125          END IF
126          CALL CURVE(DXARY(IPLT),DYARY(IPLT),NPTS,0)
127          END IF
128          END IF
129          40 CONTINUE
130          10 CONTINUE
131          CALL RESET('HEIGHT')
132          CALL RESET('XNAME')
133          CALL RESET('YNAME')
134          CALL RESET('DOT')
135          CALL ENDPL(0)
136          RETURN
137          END

```

◆PRT.S .TIMER

LAMBERT*TPFS(O).XQTSN(3)

```
1  *XQT MPO051/SIGN
2  *
3  *   CONTROL STATEMENT IMMEDIATELY AFTER *XQT STATEMENT
4  *       POS 5 - NO. INPUT UNITS (1, 2 OR 3)
5  *       POS 10 - LOWEST RUN NO. TO BE PROCESSED
6  *       POS 20 - HIGHEST RUN NO. TO BE PROCESSED
7  *TEST TNE/O/S6
8  *JUMP OUT
9  *ASG,TF PCSXXXXXXXXX.,U9S,SAVE05 . PCSXXXXXXXXX
10 *COPY,GM 9.,PCSXXXXXXXXX.
11 *COPY,GM 9.,PCSXXXXXXXXX.
12 *COPY,GM 9.,PCSXXXXXXXXX.
13 *TEST2:
14 *TEST TG/1/S6
15 *JUMP TEST3
16 *COPY,GM 10.,PCSXXXXXXXXX.
17 *COPY,GM 10.,PCSXXXXXXXXX.
18 *COPY,GM 10.,PCSXXXXXXXXX.
19 *TEST3:
20 *TEST TG/2/S6
21 *JUMP OUT
22 *COPY,GM 11.,PCSXXXXXXXXX.
23 *COPY,GM 11.,PCSXXXXXXXXX.
24 *COPY,GM 11.,PCSXXXXXXXXX.
25 *OUT:
26 *FREE,I PCSXXXXXXXXX.
27 *   EXITED FROM .XQTSN
```

*PRT,S .SGFLIP

LAMBERT*TPF\$(0).SGFLIP(61)

```

1      C
2      C      FUNCTION:
3      C      THIS ASCII FORTRAN DRIVER PROGRAM PERFORMS
4      C      (1) SIGN CHANGE OF NEGATIVE DATA PTS. (MANUAL ENTRIES)
5      C      TO POSITIVE AND (2) WRITES REFORMATTED DATA RECORD TO MASS STORAGE
6      C
7      C      DIMENSION DC(250)
8      C      COMMON/PHTHDR/RUNID,VARID,ISITID,FITIME,JDTDIG,DELTA,IPGVER
9      C      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,FRUN,LRUN,RUNS(200)
10     C      *,FIRST,LAST,ENC*4,SITE*1
11     C
12     C      READ USER PARAMETERS IN CARD IMAGE
13     C
14     C      READ(5,6)IFILE,FRUN,LRUN
15     6      FORMAT(15,9X,2(A6,4X))
16     C      DATA MAX/320/
17     C
18     C      COMPUTE RUN RANGE
19     C
20     C      FIRST=SUBSTR(FRUN,3,4)
21     C      LAST=SUBSTR(LRUN,3,4)
22     C      DECODE(4,FIRST)IFIRST
23     C      DECODE(4,LAST)ILAST
24     4      FORMAT(I4)
25     C
26     C      COMPUTE INPUT UNIT ASSIGNMENTS
27     C
28     C      ILIM=IFILE+8
29     C      IU=9
30     1      CONTINUE
31     C      PRINT 3,IU,ILIM,FRUN,LRUN
32     3      FORMAT(2I10,2(4X,A6))
33     C      NIU=-1*IU
34     C
35     C      BEGIN RUN LOOP
36     C
37     C      IRN=IABS(ILAST-IFIRST)+1
38     C      INC=IFIRST
39     C      DO 100,I=1,IRN
40     C      SUBSTR(RUNS(I),1,2)='LX'
41     C      ENCODE(17,ENC)INC
42     17     FORMAT(I4)
43     C      SUBSTR(RUNS(I),3,4)=ENC
44     C      INC=INC+1
45     C
46     C      ATTEMPT TO SET SECTOR TO HEADER FOR RUN
47     C
48     C      CALL SETSCT(IU,RUNS(I),'PHDATA',1,ISITE,NREC,IER)
49     C      IF(IER.NE.0)GO TO 100
50     C      CALL GETRCD(IU,MAX,RUNID,VARID,ISITE,IDL,DC,IER)
51     C      IF(IER.NE.0)THEN
52     C      PRINT 13,IER
53     13     FORMAT(1X,'STATUS=',03)
54     C      STOP 'HEADER ERROR'
55     C      END IF
56     C
57     C      SET SECTOR FOR TIME RECORD
58     C
59     C      VARID=' TS '

```

```

60      ENCODE(11,SITE)ISITID
61      11  FORMAT(I1)
62      SUBSTR(VARID,1,1)=SITE
63      CALL SETSCT(IU,RUNID,VARID,1,ISIT,NREC,IER)
64      IF(IER.NE.0)GO TO 110
65      C
66      C      READ NEXT RECORD
67      C
68      DO 10,K=1,100
69      CALL SECTOR(IU,ISEC)
70      CALL GETRCD(IU,MAX,RUNID,VARID,ISIT,IDL,DC,IER)
71      IF(IER.NE.0)THEN
72          IF(IER.EQ.1)THEN
73              GO TO 100
74          ELSE
75              GO TO 110
76          END IF
77      END IF
78      INEG=0
79      C
80      C      CHANGE SIGNS OF NEGATIVE NOS.
81      C
82      DO 8,M=1,250
83      IF(DC(M).LT.0)THEN
84          INEG=1
85          DC(M)=-DC(M)
86      END IF
87      8  CONTINUE
88      C
89      C      WRITE RECORD BACK TO MASS STORAGE
90      C
91      IF(INEG.NE.0)THEN
92          CALL SETADR(IU,ISEC)
93          CALL PUTRCD(NIU,RUNID,VARID,ISIT,IDL,DC,IER)
94          IF(IER.NE.0)GO TO 110
95      END IF
96      10  CONTINUE
97      C
98      C      ERROR ROUTINE
99      C
100     110 PRINT 111,RUNID,IER
101     111 FORMAT(' ERROR IN SIGN FLIP ROUTINE FOR RUN ',A6/' STATUS=',03)
102     100 CONTINUE
103     IF(IU.LT.ILIM)THEN
104         IU=IU+1
105         GO TO 1
106     END IF
107     END

```

SUBROUTINE SETSCT

a. Function: Subroutine SETSCT is an ASCII FORTRAN program designed to control the positioning of sector address for either a photo header record or a photo time record. The calling program must provide the logical unit number of the photo data file and the ASCII variable RUNID; these arguments are used to locate the desired run in the random access directory. In addition, the user provides the variable name (see Calling Sequence) and the desired occurrence of the time record as arguments. There may be more than one PDS (Photo Digitizing System) physical record/target in a run, hence it is necessary to designate which one is desired (even if there is only one PDS physical record/target).*

Entry DATSCT of subroutine DIRBFR is called in order to retrieve the sector address of the desired record. This address is used in a subsequent call to Entry SETADR of subroutine FORTIO in order to correctly position the data file.

This subroutine returns the site ID of the data file as argument ISITE. Additionally, the number of PDS physical records required per target for the desired run is returned in argument NREC.

b. Subroutines (see reference 1):

DIRBFR/Entry DATSCT, FORTIO/Entry SETADR

c. Calling Sequence:

CALL SETSCT (IUNIT, RUNID, VARID, NPHYS, ISITE, NREC, IER)

* In the event more than 250 frames of photo data were digitized for a target, an extra PDS physical record is required for each additional 250 frames.

SUBROUTINE SETSCT (Continued)

where:

<u>ARGUMENT</u>	<u>DESCRIPTION</u>	<u>TYPE</u>
IUNIT	Logical unit number of photo directory and data file	Integer
RUNID	Run number for requested sector positioning	6 char
VARID*	Variable name for requested sector positioning	6 char
NPHYS	Desired occurrence of time record (Nphys=1 unless more than 250 frames were digitized)	Integer
ISITE	Returned site ID of data file	Integer
NREC	Returned number PDS physical records required for each photo target	Integer
IER	Returned error status code (See Appendix 4e for code definitions)	Octal

* This argument identifies whether the sector is to be set for a header record (VARID='PHDATA') or a time record (VARID='1TS' for site 1).

SUBROUTINE GETRCD

a. Function: This subroutine is an ASCII FORTRAN program designed to control the execution of sequential binary reads of FORTRAN V photo (1) header records, (2) time records and (3) displacement records. GETRCD utilizes Entry READIO of subroutine FORTIO to retrieve the data record from mass storage (the data record is stored in named common IOBUFF). It then extracts the run number (RUNID) and variable name (VARID) from the data record and converts each from FIELDATA to ASCII with system function subprogram FD2ASC; these converted character variables are returned to the calling program. The site ID (ISITE) of the data record is also returned as an argument.

If the retrieved record is either a time record or a displacement record, the word length of the data array (IDL) and the data array itself (DATARY) are returned to the calling program. If the retrieved record is a header these arguments have no meaning. However, when a header record is read GETRCD calls subroutine DECHED in order to (1) convert all alphanumeric data in the header to ASCII and (2) store all header information in named common PTHDR.

The user specifies MAXLEN as the maximum word length of the data record being read from mass storage. GETRCD utilizes function subprogram EXPAND to insure that the user has specified a maximum length large enough to accommodate those sentinels and checksums which will be read from mass storage as part of the input record. If MAXLEN is too small, an appropriate error status code is returned.

b. Subroutines (see reference 1):

FORTIO/Entry READIO, FD2ASC, ASC2FD, DECHED, EXPAND

c. Calling Sequence:

Call GETRCD (IUNIT, MAXLEN, RUNID, VARID, ISITE, IDL, DATARY, IER)

SUBROUTINE GETRCD (Continued)

where:

<u>ARGUMENT</u>	<u>DESCRIPTION</u>	<u>TYPE</u>
IUNIT	Logical unit number of photo data and directory file	Integer
MAXLEN	Maximum word length of input record	Integer
RUNID	Returned run number of input record	6 char
VARID*	Returned variable name of input record	6 char
ISITE	Returned site ID	Integer
IDL	Returned word length of data array	Integer
DATARY	Returned data array	Real
IER	Returned error status code (see Appendix 4e for code definitions)	Octal

d. Common Block Usage:

COMMON/IOBUFF/IOBUFF(4100)

where:

<u>VARIABLE</u>	<u>DESCRIPTION</u>	<u>TYPE</u>
IOBUFF(4100)	Input array containing FORTRAN (FIELDATA) photo data record read from mass storage (see Figures 2c-2e for data record formats)	Integer

The contents of this array are passed as an argument to the header decoding subroutine (DECHED). Alternatively, if the photo input record is either a time record or a displacement record, the data array is extracted from IOBUFF for return to the calling program.

- * VARID='PHDATA' for a photo header record
VARID='ITS ', '2TS ' or '3TS ' for a photo time record (first character represents site ID)
See Figure 4 for possible values of VARID for a photo displacement record

SUBROUTINE PUTRCD

a. Function: Subroutine PUTRCD is an ASCII FORTRAN program designed to control the execution of sequential binary writes of FORTRAN (FIELDATA) photo (1) header records, (2) time records, and (3) displacement records. This routine provides the user with complete directory maintenance. The user may utilize PUTRCD to (1) create a photo data output file with accompanying directory or (2) perform sequential output of photo records without directory maintenance. If the user desires to perform sequential writes without building a corresponding directory, (s)he must transmit the negative of the logical unit number (of the photo data file) as the value of the argument IUNIT in the call statement.

In the event a photo header record is to be written, it is the user's responsibility to provide the ASCII version of the header in named common PTHDR (see Figure 2c for format). However, if the user utilizes input subroutine GETRCD to read an existing header record, its ASCII version is automatically stored in common. This ASCII version of the header must be converted to FIELDATA; this is accomplished by a call to subroutine ENCHED.

The user must provide PUTRCD with the (1) logical unit number of the photo data output file (IUNIT), (2) run number of the output record (RUNID), (3) variable name of the output record (VARID*), (4) site ID of the output record (ISITE), (5) word length of the output data array (IDL) and (6) output data array (DATARY). Values for items (5) and (6) must be assigned for all output records except header records. Items (1) through (4) are passed by PUTRCD to subroutine DIRCTY where the necessary directory entries are created.

Once the directory has been updated, PUTRCD uses the word length of the output array (IDL) and the output array itself (DATARY) to construct the output record. Entry WRITIO of subroutine FORTIO is then called to perform the sequential write. The output record is passed down in an argument array (IOBUFF), and is also stored in named common IOBUFF.

b. Subroutines (see reference 1):

ASC2FD, ENCHED, FORTIO/Entry WRITIO

c. Calling Sequence:

Call PUTRCD (IUNIT, RUNID, VARID, ISITE, IDL, DATARY, IER)

- * VARID='PHDATA' for a photo header record
- VARID='ITS ', '2TS ' or '3TS ' for a photo time record (first character represents site ID)
- See Figure 4 for possible values of VARID for a photo displacement record

SUBROUTINE PUTRCD (Continued)

where:

<u>ARGUMENT</u>	<u>DESCRIPTION</u>	<u>TYPE</u>
IUNIT	Logical unit number of photo data and directory file	Integer
RUNID	Run number of output data record	6 char
VARID	Variable name of output data record	6 char
ISITE	Site ID of output record	Integer
IDL*	Word length of output data array	Integer
DATARY*	Output data array	Real
IER	Returned error status code (see Appendix 4e for code definitions)	Octal

d. Common Block Usage:

(1) COMMON/PHTHDR/RUNID, VARID, ISITID, F1TIME, JOTDIG, DELTA, IPGVER

(See Figure 2c for format of named common PTHDR)

(2) COMMON/IOBUFF/IOBUFF(4100)

where:

<u>VARIABLE</u>	<u>DESCRIPTION</u>	<u>TYPE</u>
IOBUFF(4100)	Output array containing FORTRAN (FIELDATA) photo (1) header record, (2) time record or (3) displacement record to be written to core (see Figures 2c-2e for data record formats)	Integer

* This argument has no meaning for header record.

SUBROUTINE EOFPUT

a. Function: This ASCII FORTRAN program is designed to control the writing of an "end-of-file" to the end of a data file. Initially EOFPUT calls subroutine DIRCTY in order to have the output file positioned for the write. Once this is done, the "end-of-file" is written by virtue of a call to entry WRITIO of the subroutine FORTIO. Control is then returned to the calling program. In the event the user wishes to write an "end-of-file" without directory maintenance, (s)he must pass down the negative of the logical unit number (of the photo data file) as the value of the argument IUNIT.

b. Subroutines (see reference 1):

DIRCTY, FORTIO/Entry WRITIO

c. Calling Sequence:

Call EOFPUT (IUNIT, IER)

where:

<u>ARGUMENT</u>	<u>DESCRIPTION</u>	<u>TYPE</u>
IUNIT	Logical unit number of photo data and directory file	Integer
IER	Returned error status code (see Appendix 4e for code definitions)	Octal

ERROR CODE DEFINITIONS

IER (Octal) = error status code of the I/O function

IER (Octal)	Explanation
0	Normal I/O completion
1	"End-of-file" has been identified
5	Attempted to read from unassigned area of mass storage
20	Write attempted on read only file
21	Reference made to unassigned file
22	Write attempted beyond assigned area of mass storage
25	Maximum length of input/output array exceeded
26	FORTTRAN sentinels are unequal - read was not executed
30	Run number not found in run directory
31	Variable name not found in variable directory
32	Checksum error
33	Run number not last entry in run directory
34	Update in place aborted due to different data record lengths
35	Invalid FORTTRAN sentinel - word length indicated for physical record greater than 249
36	Invalid FORTTRAN sentinel - record number indicated for first physical record not equal to 1

END

FILMED

9-84

DTIC